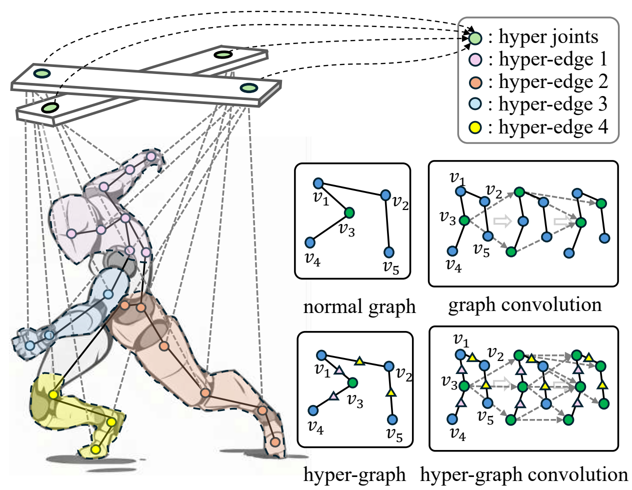
Adaptive Hyper-Graph Convolution Network for Skeleton-based Human Action Recognition with Virtual Connections

Youwei Zhou[1], Tianyang Xu[1]*, Cong Wu[1], Xiaojun Wu[1], Josef Kittler[2]
[1]Jiangnan University, China [2]University of Surrey, UK
youwei_zhou@stu.jiangnan.edu.cn
{tianyang_xu, congwu, wu_xiaojun}@jiangnan.edu.cn
j.kittler@surrey.ac.uk

Abstract

The shared topology of human skeletons motivated the recent investigation of graph convolutional network (GCN) solutions for action recognition. However, most of the existing GCNs rely on the binary connection of two neighboring vertices (joints) formed by an edge (bone), overlooking the potential of constructing multi-vertex convolution structures. Although some studies have attempted to utilize hyper-graphs to represent the topology, they rely on a fixed construction strategy, which limits their adaptivity in uncovering the intricate latent relationships within the action. In this paper, we address this oversight and explore the merits of an adaptive hyper-graph convolutional network (Hyper-GCN) to achieve the aggregation of rich semantic information conveyed by skeleton vertices. In particular, our Hyper-GCN adaptively optimises the hyper-graphs during training, revealing the action-driven multi-vertex relations. Besides, virtual connections are often designed to support efficient feature aggregation, implicitly extending the spectrum of dependencies within the skeleton. By injecting virtual connections into hyper-graphs, the semantic clues of diverse action categories can be highlighted. The results of experiments conducted on the NTU-60, NTU-120, and NW-UCLA datasets demonstrate the merits of our Hyper-GCN, compared to the state-of-the-art methods. The code is available at https://github.com/6UOOON9/Hyper-GCN.



(a) (b)

Figure 1. **Illustration of our Hyper-GCN.** In (a), dotted lines represent the virtual connections. Each coloured part containing multiple joints represents a hyper-graph with hyper-edges. (b) provides an intuitive comparison between normal graph convolution and hyper-graph convolution operations with the same connectivity degree.

1. Introduction

Skeleton-based human action recognition is a popular research topic in artificial intelligence, with practical applications in video understanding, video surveillance, human-computer interaction, robot vision, VR and AR [16, 21, 28–30, 42, 49, 49]. In general, a skeleton sequence contains a series of 2D or 3D coordinates, which can easily be collected by low-cost depth sensors or obtained by video-based pose estimation algorithms [26, 34, 35, 43]. Compared to RGB and optical flow images, skeleton data, which represents the basic physical structure of a human being, is of lower dimension, conveying human action with higher efficiency. Moreover, it is robust to illumination changes and scene variations. For these reasons, the adoption of this structural data is very popular in skeleton-based action recognition [3, 6, 10, 25, 35, 45, 51].

To facilitate skeleton-based action recognition, both Recurrent Neural Networks (RNNs) [10, 25, 35, 45] and Con-

*Corresponding author

volutional Neural Networks (CNNs) [7, 17, 40] have been well explored. However, RNNs themselves cannot depict the intrinsic skeleton topology. The learned filters of CNNs, on the other hand, neglect the spatio-temporal structure of the skeleton. Drawing on these observations, recent studies have focused on how to model the skeleton topology directly. In principle, the physical topology of human joints and bones can be consistently represented by a graph. Accordingly, the graph convolutional network (GCN) is typically introduced to aggregate feature information conveyed by skeleton joints [23, 52, 57]. In terms of data representations, skeleton joints are also one-dimensional sequence data, similar to language. Therefore, with the rise of recent self-attention techniques [9, 44], several Transformer-based action recognition solutions [32, 38, 47, 53] have been released. Transformer considers skeletal joints as tokens and uses attention maps to reflect the topology. These methods have achieved excellent performance but sacrifice much higher GFLOPS and parameters than GCN.

In general, two neighbouring joints can pass messages through their shared bone. In graph terms, this corresponds to the exchange of information between two vertices along their connecting edge. Besides the physical skeleton topology, attempts have been made to explore the implicit relationships among joints [3, 6, 23, 36], suggesting different variants of skeleton topology. In principle, these approaches assume a binary connection between each connected vertex pair. Mathematically, the constructed topology in the adjacency matrix is represented by a normal graph. However, human actions are jointly defined by several joints. Hence, human actions encompass not only binary relations between vertex pairs but also multi-joint relationships. For instance, the action primitive *starting running* is manifest in the human raising the left hand while the right leg steps forward, as shown in Figure 1 (a). The binary connections are not sufficient to capture the synergistic interaction of multiple joints. This strongly argues for constructing feature aggregation paths involving multiple vertices.

Accordingly, we propose to construct a hyper-graph to depict the skeleton topology and take advantage of the outstanding performance of hyper-graph analysis techniques [2, 12, 18, 19]. The hyper-graph topology involves multiple node connections, rather than binary connections of a normal graph. As shown in Figure 1 (a), in a hyper-graph, a hyper-edge can link more than two vertices. This linking has the capacity to represent complex collaborative relations among human joints. As one hyper-edge associates multiple vertices, a single hyper-graph convolution enables aggregating all the features along the hyper-edges.

An illustration is provided in Figure 1 (b), where the normal graph and hyper-graph are presented to demonstrate their differences in passing the information conveyed by the vertex features. In the normal graph convolution, after 2-layer aggregation, the information of the green vertex is spread to 2 other vertices. In contrast, in a hyper-graph convolution, the information of the green vertex spreads to all the vertices, creating an extended receptive field. Theoretically, by modelling a skeleton using a hyper-graph for the aggregation of joint information, improved efficiency can be obtained during learning the action semantics.

However, the key to achieving enhanced information aggregation by hyper-graph convolution lies in the conformity of the hyper-graph structure to the real topology of human action. Several studies [13, 58] attempt to adopt a hyper-graph to model the skeleton topology, mostly based on superficial prior knowledge and an unfounded definition of the structure of hyper-graph. Distinguishing our approach from the existing hyper-graph-based methods, we design an adaptive solution for constructing the hyper-graph with virtual connections. It can construct a Non-uniform hyper-graph specific to each individual action type.

Besides the information aggregation issue, the capacity of an action recognition system is also constrained by its input features. In general, the number of skeleton joints is fixed in existing benchmarks, *e.g.*, 25 for NTU-120 [26]. The entire recognition process relies on a successful capture of the interactions among the skeleton joints. In the domain of artistic puppetry, the history of driving actions, such as Shadow Play [1] and Marionette [2] goes back more than 2000 years. This kind of art form provides an inspiration for involving additional 'hyper joints' which can drive or facilitate a better communication between existing joints. The underlying spirit is to alleviate the pressure on the real joints to store and transfer complex semantics. Jointly with the hyper joints, real joints can focus more on storing neighbouring joint features and relegating the task of transferring the global information to the hyper joints. Interestingly, the class token in existing Transformers can also be considered as a hyper-token [9, 44]. As shown in Figure 1, the skeleton is described as a marionette, where the actions are "controlled" by connecting hyper joints to the real joints. This suggests that the hyper joints are not only able to capture the representation information of human action, but also reveal the implicit information between physically connected joints as hints for recognition.

By endowing an adaptive non-uniform hyper-graph with hyper joints, virtual connections are created to perform comprehensive hyper-graph convolutions. We construct Hyper-GCN based on the above design principles. Extensive experiments on 3 datasets, NTU RGB+D 60 [35], NTU RGB+D 120 [26], and NW-UCLA [46], are conducted for evaluation. The results validate the merits of our proposed Hyper-GCN. The main contributions are as follows:

- An adaptive non-uniform hyper-graph to represent the hu-

[1] https://en.wikipedia.org/wiki/Shadow_play
[2] https://en.wikipedia.org/wiki/Marionette

man skeleton topology. Compared with a fixed hyper-graph, the constructed topology is more action-specific, thus boosting the discrimination.
- The injection of virtual hyper joints, enriching the connectivity of the physical joints, from a global semantic perspective.
- As the processing architecture, we propose a hyper-graph convolution network (Hyper-GCN). The SOTA performance achieved on three public datasets demonstrates the merits of the Hyper-GCN and virtual connection designs.

2. Related Work

2.1. Graph Topology for Action Recognition

A graph can represent a human skeleton, preserving the joint relationships via predefined edges. To aggregate semantics, GCNs [1, 4, 11, 20, 24, 37, 54, 57] and Transformers [8, 32, 38, 53, 55] have been well studied.

For GCN-based approaches, ST-GCN [52] proposes to represent the topology with an adjacency matrix of 3 subsets by modeling the spatio-temporal relevance. Similarly, 2s-AGCN [36], InfoGCN [6], and DS-GCN [51] use a self-attentive mechanism to learn the topology from joint pairs. Besides the intuitive spatio-temporal dimensions, CTR-GCN [3] proposes to learn the topology for channels to refine the skeleton.

For Transformer-based solutions, IIP-Transformer [47] adopt a self-attention mechanism to establish the intra-part and the inter-part joints. IGFormer [31] learns the topology between persons at both semantic and distance levels. STTFormer [33] proposed structure can capture the correlation between joints in consecutive frames.

All the above two paradigms construct the topology from joint pairs, which rely only on the binary relation between two vertices. In their implementation stage, the adjacency matrix or attention map is used to represent the topology. In this case, high-order information among joints is not taken into consideration, neglecting the collaborative power among multiple joints. Though hyper-graph is considered to construct topology using multivariate joint relationships, current [13, 58] solutions manually set the hyper-graphs, which greatly relies on human experience, sacrificing the adaptability of graph learning.

2.2. Feature Configurations for Action Recognition

It has been observed that the input features play the most essential role in delivering high recognition accuracy [1, 5, 15, 27, 31, 33, 41, 47] in GCNs and Transformers.

Drawing on this, Graph2Net [50] proposes to extract local and global spatial features jointly. CTR-GCN [3] uses multi-scale temporal convolution to extract temporal features. While HD-GCN [23] introduce the hierarchical edge convolution to key edge features. To extend the perception field, STC-Net [22] uses the dilated kernels for graph convolution to capture the features.

Similarly, IG-Transformer [31] aggregate the features of two persons as one to mine the semantic information. STST [55] split the coordinate features, semantic features and temporal features into three kinds of tokens. TSTE [53] extracting and merging the spatial and temporal features as input to transformer encoders.

After all, existing methods [6, 14, 32, 36, 38, 51, 52] receive the input features from real skeleton joints. However, each skeleton joint acts as an information carrier during forward passing, which is required to deliver both local context and global semantics. Given a fixed representative capacity (number of channels), we believe it is necessary to involve additional virtual joints to balance the pressure of storing local and global information.

3. Approach

3.1. Preliminaries

In general, the input features in graph convolution are represented by $\mathbf{F_{in}} \in \mathbb{R}^{C\times T\times V}$, where C represents the number of channels in the feature maps. Given the topology of a human skeleton, we usually define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of joints and $\mathcal{E}$ represents the set of edges between joints. For the set of edges $\mathcal{E}$, it is formulated as an adjacent matrix $A \in \mathbb{R}^{N\times N}$, where N represents the number of human joints. The normalised adjacency matrix is represented by $\hat{A} \in \mathbb{R}^{N\times N}$. The normalisation operation is formulated as follows:

$$\hat{A} = \Lambda^{-\frac{1}{2}} A \Lambda^{-\frac{1}{2}}, \tag{1}$$

where Λ represents the diagonal matrix stored with the degree of every joint. The entire normal graph convolution process can be formulated as:

$$\mathbf{F_{out}} = \sigma(\hat{A}\mathbf{F_{in}}P), \tag{2}$$

where $P \in \mathbb{R}^{C\times C'}$ is the learnable parameters, representing the feature transformation patterns in the feature space. σ denotes the non-linear activation function ReLU.

3.2. Hyper-graph

Here, we use $\mathcal{G}_\mathcal{H} = (\mathcal{V}_\mathcal{H}, \mathcal{E}_\mathcal{H}, \mathcal{W}_\mathcal{H})$ to define the spatial hyper-graph [12] with human skeleton. $\mathcal{V}_\mathcal{H}$ and $\mathcal{E}_\mathcal{H}$ follow similar definitions in the normal graph, which represent the set of joints and the set of hyper-edges. In addition, we introduce $\mathcal{W}_\mathcal{H}$ to represent the weights of each hyper-edge. Since a hyper-edge contains multiple joints, the corresponding topology can no longer be simply represented by an adjacency matrix. Therefore, we introduce the incidence matrix $H \in \mathbb{R}^{N\times M}$ to describe the topology of each joint in the hyper-graph. N represents the number of joints and M

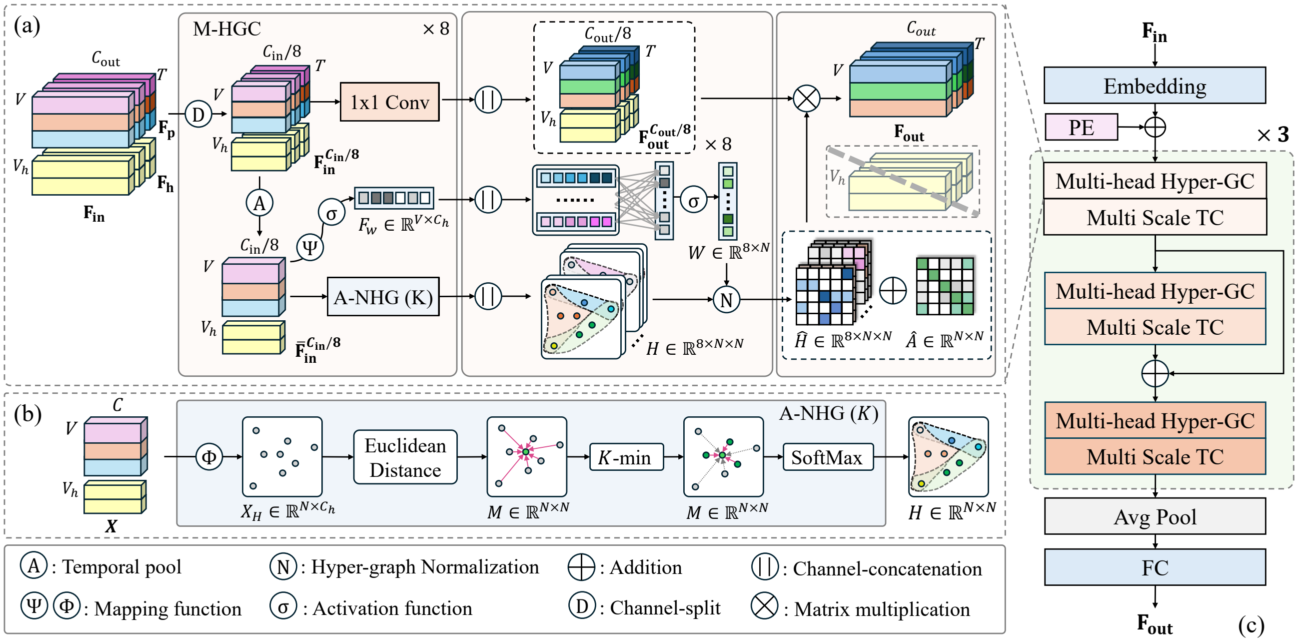


Figure 2. **The framework of our Hyper-GCN.** Part (a) represents the Multi-head Hyper-graph Convolution (M-HGC) module. Part (b) represents the process of constructing an adaptive hyper-graph. Part (c) represents the architecture of Hyper-GCN. $\mathbf{F}_\mathbf{p}$ represents the features of real joints. $\mathbf{F}_h$ represents the hyper-joints which are learnable parameters. $\hat{A}$ represents the physical topology.

represents the number of hyper-edges. Given $v \in \mathcal{V}_\mathcal{H}$ and $e \in \mathcal{E}_\mathcal{H}$, the values of incidence matrix be determined by:

$$h(e, v) = \begin{cases} 1, & v \in e, \\ 0, & v \notin e \end{cases}. \tag{3}$$

Similar to the normal graph convolution, it also needs to normalise the hyper-graph to modulate the aggregated features. The degree of hyper-graph consists of the degree of joints and the degree of hyper-edges. The degree of joints is represented by the sum of weights of all joints contained in each hyper-edge. Given $v \in \mathcal{V}_\mathcal{H}$, it can be described as follows:

$$d(v) = \sum_{e \in \mathcal{E}_\mathcal{H}} w(e) h(v, e), \tag{4}$$

where we use a diagonal matrix $D_v \in \mathbb{R}^{N \times N}$ to represent the degree of joints. $W \in \mathbb{R}^{M \times M}$ represents the weight matrix of hyper-edges, which is formulated as a diagonal matrix. The degree of hyper-edges represents the sum of the number of joints contained in each hyper-edge. Given $e \in \mathcal{E}_\mathcal{H}$, it can be described as follows:

$$d(e) = \sum_{v \in \mathcal{V}_\mathcal{H}} h(v, e), \tag{5}$$

where we use a diagonal matrix $D_e \in \mathbb{R}^{M \times M}$ to represent the degree of hyper-edges. Therefore, the normalisation of a hyper-graph is formulated as follows:

$$\hat{H} = D_v^{-1} H W D_e^{-1} H^T, \tag{6}$$

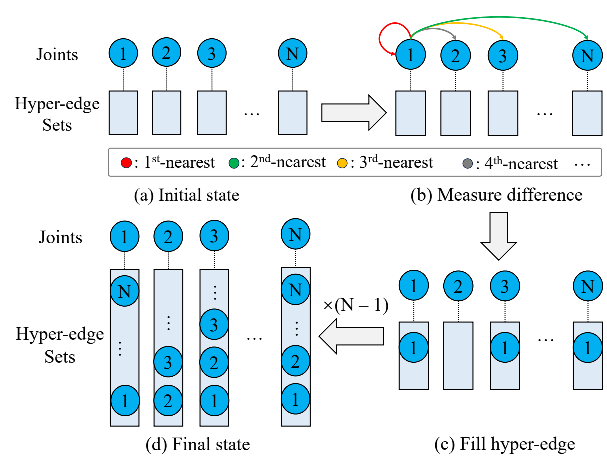


Figure 3. **Instance of the incidence matrix construction by A-NHG with hyper-parameter $K = 3$.** N represents the number of joints. (a) represents the hyper-edge sets are null. (b) represents that measure the distance between 1-st joint and other joints. (c) represents that retrain K-nearest hyper-edges which contain 1-st joint. (d) represents the final state.

where $\hat{H} \in \mathbb{R}^{N \times N}$ represents the normalised incidence matrix for hyper-graph convolution.

3.3. Adaptive Non-uniform Hyper-graph

Constructing the hyper-graph incidence matrix is critical to understanding human actions. As shown in Figure 2 (b), we design an Adaptive Non-uniform Hyper-graph (A-NHG) Construction.

Given the general features $\mathbf{X} \in \mathbb{R}^{N\times C}$ of spatial skeleton joints, N represent the number of joints, C represent the channels. Our A-NHG construct the hyper-edge with each joint as the center of mass, and a total of N hyper-edges are constructed. A-NHG utilises the Euclidean distance to measure the joint difference.

We use one mapping function $\Phi \in \mathbb{R}^{C\times C_h}$ to embed the features into the subspace for constructing the hyper-graph. C_h represents the hidden channels of the mapping subspace. This design enables preserving the original features to reflect their spatial relevance. Based on $X_H \in \mathbb{R}^{N\times C_h}$, we define the set of joints $\mathcal{V_H} = \{v_1 \in \mathbb{R}^{C_h}, v_2 \in \mathbb{R}^{C_h}, ..., v_N \in \mathbb{R}^{C_h}\}$. Given $v_i, v_j \in \mathcal{V_H}$, each element $m_{i,j}$ in the distance matrix $M \in \mathbb{R}^{N\times N}$ can be obtained as follows:

$$m_{i,j} = m_{j,i} = ||v_i - v_j||_2. \tag{7}$$

In principle, we need to transform the distance matrix into a probability incidence matrix, which determines whether the specific joints are contained with the same hyper-edges. In order to guarantee that the model is trainable, we use the softmax operation to assign the probabilities. However, directly using the softmax would inevitably result in each joint belonging to all hyper-edges to some extent.

Since this outcome is undesirable, we introduce constraints on each joint to limit the number of hyper-edges it can belong to, preventing excessive connections. For one joint, we retain only the K nearest hyper-edges represented by joints in the semantic space for probability estimation. And K is the hyper-parameter of A-NHG.

Given the row vector $\mathbf{m}_i$ in the distance matrix M, which represents the distances between joint i and the other joints. We select the indices of the minimum K joints from $\mathbf{m}_i$ and add them to the set set_i. Based on set_i, $h_{i,j}$ in the incidence matrix $H \in \mathbb{R}^{N\times N}$, can be calculated as follows:

$$h_{i,j} = \begin{cases} \frac{\exp(-m_{i,j})}{\sum_{k\in set_i} \exp(-m_{i,k})}, & j \in set_i, \\ 0, & j \notin set_i. \end{cases} \tag{8}$$

In contrast to uniform hyper-graph (every hyper-edge contains the same number of joints), our approach limits the maximum number of every joint that can be contained by hyper-edges. Obviously, the number of joints contained in each hyper-edge constructed in this way is non-fixed, allowing the hyper-edges to capture more differentiated associations. The entire process for example with K = 3 is illustrated in the Figure 3.

3.4. Multi-head Hyper-graph Convolution

To accommodate the semantic information reflected by different group channels, we further propose the Multi-head Hyper-graph Convolution (M-HGC), as shown in Figure 2 (a).

We use separate branches to independently perform hyper-graph convolution on the topologies represented by the multi-head hyper-graphs, thereby enhancing the computational efficiency of the Hyper-GCN. For instance, the features $\mathbf{F_{in}} \in \mathbb{R}^{C_{\text{in}}\times T\times V}$ are divided into 8 separate branches along the channel dimension, which are processed by 8 M-HGC parallelly, delivering 8 separate hyper-graphs.

Next, we perform temporal pooling, and then the obtained spatial information is the evidence to determine the optimal hyper-graph construction. This can efficiently decouple the temporal and spatial clues. After performing temporal average-pooling, we can obtain $\bar{\mathbf{F}}_{\mathbf{in}} \in \mathbb{R}^{C_{\text{in}}\times V}$.

The hyper-graph is depicted by the incidence matrix and weight matrix. So, we use a separate mapping function for each head to embed the features into the subspace for constructing the hyper-graph. This design enables preserving the original features to reflect their spatial relevance. After that, we introduce the A-NHG to obtain the incidence matrix. For the weight matrix, we adopt MLP to measure each hyper-edge. The hyper-graph obtained operation can be formulated as follows:

$$H = \uplus_{k=1}^{8} \text{A-NHG}(\bar{\mathbf{F}}_{\mathbf{in}}^k), \tag{9}$$

$$W = \sigma_2(\Psi_2(\uplus_{k=1}^{8} \sigma_1(\Psi_1^k(\bar{\mathbf{F}}_{\mathbf{in}}^k)))), \tag{10}$$

where $\Psi_1 \in \mathbb{R}^{C_{\text{in}}/8\times C_h}, \Psi_2 \in \mathbb{R}^{(8\times C_h)\times 8}$ are the mapping functions, which are set as learnable parameters. C_h represents the hidden channels of the mapping subspace. k represents the channels $[(k-1)\times C_{\text{in}}/8+1, ..., k\times C_{\text{in}}/8]$ of $\bar{\mathbf{F}}_{\mathbf{in}}$. $\uplus$ represents the channel concatenation. σ_1, σ_2 denote the activation function LeakeyReLU and Tanh to obtain the weight of each hyper-edge, limiting the values with the range of $[-1, 1]$. Based on $H \in \mathbb{R}^{8\times N\times N}, W \in \mathbb{R}^{8\times N}$, normalise the hyper-graph by sec.3.2 to obtain the $\hat{H} \in \mathbb{R}^{8\times N\times N}$.

Besides the hyper-graph, we incorporate the physical topology to emphasise the natural physical relations of a human being's skeleton. To achieve this objective, existing methods [3, 23, 36, 52] divide the physical topology of the human body into 3 subsets. They are represented by $S = \{s_{id}, s_{cf}, s_{cp}\}$, where s_{id}, s_{cf} and s_{cp} denote the identity, centrifugal, and centripetal joint subsets. In our M-HGC, to ensure that the integrated physical topology in each head remains complete, we merge them into a single set. 8 heads are aggregated by channel concatenation after the hyper-graph convolutions, as shown in Figure 2 (a). The operation of M-HGC can be formulated as follows:

$$\mathbf{F_{out}} = \uplus_{k=1}^{8} (\hat{A}_k + \alpha \cdot \hat{H}_k)\mathbf{F}_{\mathbf{in}}^k P_k, \tag{11}$$

where $\hat{A}$ and $\hat{H}$ denote the normalized physical adjacency matrix and normalized incidence matrix. P is the learnable weight parameter for feature transform. α is a learnable parameter for the topology fusion of each head.

3.5. Virtual Connections

It is worth emphasising that incorporating learnable joints among different samples is essential for enhancing the model capacity, as these learnable joints capture generalised features of human actions. This not only enriches the semantic information but also facilitates easier interaction connections among real joints. Therefore, we introduce the hyper-joints which are to participate in the hyper-graph convolution as shown in Figure 2 (a).

The shape of hyper-joints is consistent with the features of physical joints $\mathbf{F_p} \in \mathbb{R}^{C \times T \times V}$, which is described as $\mathbf{F_h} \in \mathbb{R}^{C \times T \times V_h}$ (V_h represents the number of hyper-joints). To ensure alignment in the temporal dimension, the hyper-joints are shared across each frame. Specifically, the learning of hyper-joints is supervised by the loss function, which guides these hyper-joints to support generalisable driven features embedded within large amounts of data. Furthermore, we set independent hyper-joints at each layer of Hyper-GCN, aiming to harmonise features at different depths. Typically, these hyper-joints are involved in spatial hyper-graph convolution rather than in temporal convolution.

To diversify these hyper-joints, we propose the Divergence Loss for hyper-joints optimisation to mitigate their homogenisation. We adopt a cosine matrix $C \in \mathbb{R}^{V_h \times V_h}$ to measure the differences between hyper-joints, which can be formulated as follows:

$$C = \frac{\mathbf{F_h}\mathbf{F_h}^T}{||\mathbf{F_h}||^2}, \tag{12}$$

Specifically, in Divergence Loss, we measure the differences of hyper-joints with the mean of the cosine matrix C in each layer. Since the correlation of hyper-joints themselves cannot be optimised, we manually eliminate this part by minus the V_h. The loss calculation can be formulated as:

$$\mathcal{L}_h(C) = \frac{\sum_{i=1}^{V_h}\sum_{j=1}^{V_h} ReLU(c_{i,j}) - V_h}{V_h(V_h-1)}, \quad c_{i,j} \in C, \tag{13}$$

$$\mathcal{L} = \mathcal{L}_{CE} + \frac{1}{L}\sum_{l=1}^{L} L_h(C_l), \tag{14}$$

where V_h represents the number of hyper-joints we introduced. C_l represents the cosine matrix of the l-th layer. $\mathcal{L}_{\text{CE}}$ represents the cross-entropy loss. Additionally, we manually connect the hyper joints to all the physical joints, which optimises the topology from the datasets.

3.6. Entire Architecture

Hyper-GCN consists of an embedding layer [6], 9 spatial-temporal convolution layers as shown in Figure 2 (c). Each layer consists of the proposed M-HGC and Multi Scale Temporal convolution (MS-TC) [3]. 9 Layers are categorised into 3 stages. In each stage, we introduce the dense connections to integrate deep and shallow features in each stage. Additionally, to further validate the potential of Hyper-GCN, we propose a base version and a large version. The channels in each stage are set to 128, 256, 256 for the base version and 128, 256, 512 for the large version.

4. Evaluation

4.1. Datasets

NTU-RGB+D 60 & 120 NTU-RGB+D 60 [35] is a large dataset widely used in skeleton-based human action recognition, which is categorised into 60 classes. NTU-RGB+D 120 [26] is an extention to 120 classes of NTU 60. 4 benchmarks recommended by the official are adopted: (1) NTU60-XSub, (2) NTU60-XView, (3) NTU120-XSub, (4) NTU120-Xset.

Northwestern-UCLA. The Northwestern-UCLA (NW-UCLA) dataset [46] contains 1494 video clips, which is categorised into 10 classes. It contains 3 different camera views and is performed by 10 actors.

4.2. Implementation Details

Our training and evaluation stages are on a single GPU RTX 3090. In the training phase, Hyper-GCN is optimised by Stochastic Gradient Descent (SGD) with Nesterov momentum set at 0.9 and a weight decay at 0.0004. Our implementation uses label smooth cross-entropy loss with the Divergence Loss we proposed. We set a total of 140 epochs with the start 5 warm-up epochs. The initial learning rate is 0.05, which is reduced to 0.005 at epoch 110 and to 0.0005 at epoch 120.

4.3. Comparison with the State-of-the-Art

Multi-stream ensemble proposed in [39] have been proven effective by most of the existing state-of-the-art methods. We also use a 4-stream ensemble to evaluate the base version and the large version of Hyper-GCN. The detailed results are reported in Table 1.

In summary, the base version of Hyper-GCN comprehensively outperforms all GCN-based and HGCN-based (hyper-graph utilised) SOTA and surpasses the Transformer-based SOTA on the NTU120, with a most lightweight design. Moreover, when scaled to match the parameters of the lightest Transformer-based method, the large version of Hyper-GCN achieves 1-st place in 4 benchmarks and 2-nd place in 1 benchmark. This demonstrates the effectiveness and merits of Hyper-GCN.

Category	Methods	Modalities	Params (M)	GFLOPs	NTU-RGB+D 60		NTU-RGB+D 120		NW-UCLA (%)
					X-Sub (%)	X-View (%)	X-Sub (%)	X-Set (%)	
GCN	ST-GCN [52]	J+B	-	-	81.5	88.3	70.7	73.2	-
	2s-AGCN [36]	J+B	-	-	88.5	95.1	82.5	84.2	-
	DC-GCN+ADG [4]	J+B+JM+BM	4.9	1.83	90.8	96.6	86.5	88.1	95.3
	MS-G3D [27]	J+B+JM+BM	2.8	5.22	91.5	96.2	86.9	88.4	-
	MST-GCN [11]	J+B+JM+BM	12.0	-	91.5	96.6	87.5	88.8	-
	CTR-GCN [3]	J+B+JM+BM	1.5	1.97	92.4	96.4	88.9	90.6	96.5
	EfficientGCN-B4 [41]	J+B+JM+BM	2.0	15.20	91.7	95.7	88.3	89.1	-
	InfoGCN [6]	J+B+JM+BM	1.6	1.84	92.7	96.9	89.4	90.7	96.6
	FR Head [56]	J+B+JM+BM	2.0	-	92.8	96.8	89.5	90.9	96.8
	HD-GCN* [23]	J+B+J'+B'	1.7	1.77	93.0	97.0	89.8	91.2	96.9
	DS-GCN [51]	J+B+JM+BM	-	-	93.1	97.5	89.2	90.3	-
	BlockGCN [57]	J+B+JM+BM	1.3	**1.63**	93.1	97.0	90.3	91.5	96.9
Transformer	DSTA-Net [38]	J+B+JM+BM	3.5	16.18	89.5	95.7	86.6	89.0	-
	IIP-Transformer [47]	J+B+JM+BM	2.9	7.20	89.5	95.7	89.9	90.9	-
	SkateFormer [8]	J+B+JM+BM	2.0	3.62	93.5	**97.8**	89.8	91.4	**98.3**
HGCN	Hyper-GNN [13]	J+B+JM+BM	-	-	89.5	95.7	-	-	-
	Selective-HCN [58]	J+B+JM+BM	-	-	90.8	96.6	-	-	-
	DST-HCN [48]	J+B+JM+BM	3.5	2.93	92.3	96.8	88.8	90.7	96.6
	Ours (B)	J+B+JM+BM	**1.1**	**1.63**	93.3	97.4	90.5	91.7	97.2
	Ours (L)	J+B+JM+BM	2.3	2.88	**93.7**	**97.8**	**90.9**	**92.0**	97.6

Table 1. **Comparison of Hyper-GCN with advanced solutions on NTU-RGB+D 60, NTU-RGB+D 120, and NW-UCLA datasets.** For a fair comparison, we use a 4-streams ensemble to evaluate the performance of the mentioned methods. J, B, JB, and JM represent the joint, bone, joint motion, and bone motion. In addition, for HD-GCN [23], we use the results from 4-streams according to the original papers for a fair comparison. The **bold** font represents the best result and underline font represents the 2-nd best result.

	K	NTU RGB+D 120 X-Sub (%) Uniform	Non-uniform
Baseline	-	84.7	84.7
w M-HGC	3	86.4 (↑ 1.7)	86.0 (↑ 1.3)
	5	86.5 (↑ 1.9)	86.2 (↑ 1.5)
	7	86.3 (↑ 1.6)	86.5 (↑ 1.8)
	9	86.0 (↑ 1.3)	**86.7 (↑ 2.2)**
	11	85.9 (↑ 1.2)	86.4 (↑ 1.7)

Table 2. **Ablation of the hyper-parameter K in A-NHG.** The Uniform represents K-uniform hyper-graph, in which one hyper-edge contains K joints. The non-uniform represents non-uniform hyper-graph, in which one joint is contained by K hyper-edge.

4.4. Ablation Study

We conduct ablation experiments with visualisations to analyse the effectiveness of our design. **red**All the ablation study is evaluated with the joint modality of base version.

Hyper-graph Construction: The Non-uniform hyper-graphs with hyper-parameter K represent the maximum number of hyper-edges contained in each joint. They reflect the varying degrees of aggregation in the constructed topology. In general, it is common to adopt a uniform hyper-graph with hyper-parameter K. Therefore, to explore how to set the hyper-parameter K for the adaptive hyper-graph in M-HGC, we design the ablation experiments to investigate the impact of hyper-parameter K on the model's recognition ability, where both uniform and non-uniform hyper-graphs are considered.

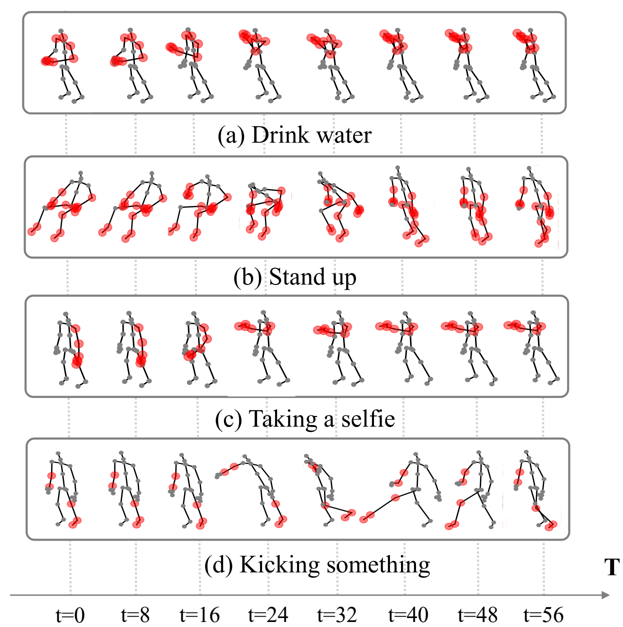


Figure 4. **Visualisation of hyper-graph in selected actions.** For ease of observation, the joints contained in the selected hyper-edge are highlighted in red.

The results are reported in Table 2. In both uniform and non-uniform hyper-graphs, a very large K means that more joints are contained in each hyper-edge. Even though these "extra" joints may be optimised for minimal weight, they are likely to act as "noise", disrupting the information inter-

		NTU RGB+D 120 X-Sub (%)	
		w/o $\mathcal{L}_h$	w/ $\mathcal{L}_h$
Baseline		84.7	84.7
w/o M-HGC	w/ 1	84.9 ($\uparrow$ 0.2)	84.9 ($\uparrow$ 0.2)
	w/ 3	84.9 ($\uparrow$ 0.2)	85.2 ($\uparrow$ 0.5)
	w/ 5	84.7	85.0 ($\uparrow$ 0.3)
w M-HGC	w/ 1	86.7 ($\uparrow$ 2.0)	86.7 ($\uparrow$ 2.0)
	w/ 3	86.6 ($\uparrow$ 1.8)	**86.9** ($\uparrow$ 2.2)
	w/ 5	86.6 ($\uparrow$ 1.8)	86.8 ($\uparrow$ 2.1)

Table 3. **Ablations on hyper joints.** w/ N represents the number of hyper joints in each layer. w/ $\mathcal{L}_h$ represents the training with Divergence Loss. The hyper-parameter K of M-HGC is set to 9.

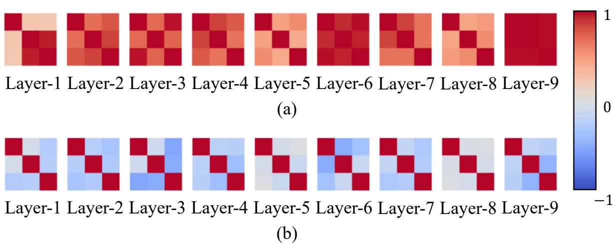


Figure 5. **Comparison of the cosine matrices of hyper-joints.** Part (a) and (b) represents w/o Divergence Loss and w/ Divergence Loss with 3 hyper-joints. It is calculated by Eqn. (12).

action represented by the hyper-edge. As the K decreases, it is difficult for the hyper-edge to represent complex joint combination patterns. Therefore, setting $K = 5$ in uniform and $K = 9$ in Non-uniform achieves the best performance. The non-uniform hyper-graph of A-NHG outperforms the uniform hyper-graph, as the combination of skeleton joints in different actions is diverse.

Furthermore, we visualise the hyper-graph inferred by Hyper-GCN on selected actions in Figure 4. The hyper-graph construction focuses on the joints that are most relevant to the corresponding action category. For “Kicking something” as part (d), the left leg joints and the right hand joints are modelled as a hyper-edge. Because in this action, the person’s right hand is previously extended forward, and the left leg swings relatively to the right hand. Similarly, in the action ”Stand up” as shown in part (b), the hand and foot joints serve as fixed joints, while the remaining body joints move relative to them. In addition, ”Taking a selfie” and ”Drinking water” both focus on the main joints of movement with the above view as Figure 4 (a), (c).

Virtual Connections: We conduct the ablation study to analyse the hyper joints and the Divergence Loss. The results are listed in Table 3. By comparison, the performance is not monotonically increasing with the number of hyper joints. The case of introducing only 3 hyper joints achieves the best performance. As our hyper joints are learned from a large amount of data, involving a large number of hyper joints can introduce redundant and ambiguous clues, degrading the performance.

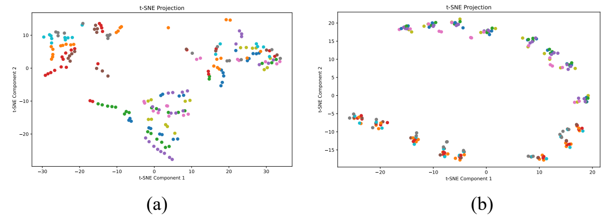


Figure 6. **t-SNE visualisation of Baseline and Hyper-GCN.** Different colours represent different joints. Part (a) and (b) reprsent the output features in Baseline and Hyper-GCN.

In addition, we visualise the cosine matrices of Hyper-GCN, as shown in Figure 5. Clearly, in the absence of the Divergence Loss, the homogenisation of the learned hyper-joints is severe. This impedes the ability of hyper-joints to represent the generalised features of human actions. The observation further validates the effectiveness of the Divergence Loss for hyper-joint optimisation, supporting our viewpoint that generalised features with a certain degree of differentiation are needed to participate in hyper-graph convolution.

Effectiveness of Hyper-GCN: To further validate that Hyper-GCN increases the efficiency of information interaction, we perform t-SNE projections on the output features in last layer between Baseline and Hyper-GCN, as shown in Figure 6 for comparison. Notice that the semantic information represented by the last Hyper-GCN layer is very convergent. This represents that the semantic information of each joint is adequately conveyed in M-HGC. The joints with similar semantics can prevent sacrificing the information after global average pooling.

5. Conclusion

In this paper, we propose an Adaptive Hyper-graph Convolutional Network for skeleton-based human action recognition. To exploit the implicit topology of multivariate synergy between joints, we introduce the Adaptive Non-uniform Hyper-graph, the Multi-head Hyper-graph Convolution and virtual connections. We carry out experiments on the dataset NTU-RGB+D 60 & 120, NW-UCLA to validate the effectiveness of Hyper-GCN. The experimental analysis verifies that our design can improve the recognition performance. To the best of our knowledge, Hyper-GCN achieves the SOTA performance on 3 public datasets. The involvement of adaptive non-uniform hyper-graph modelling indeed extends existing GCN-based action recognition paradigms.

Acknowledgment This work was supported in part by the National Natural Science Foundation of China (62020106012, 62106089, 62332008, 62336004), the Engineering and Physical Sciences Research Council (EPSRC), U.K. (Grants EP/V002856/1, and EP/T022205/1), and the Fundamental Research Funds for the Central Universities (JUSRP202504007).

References

[1] Tasweer Ahmad, Lianwen Jin, Luojun Lin, and GuoZhi Tang. Skeleton-based action recognition using sparse spatio-temporal gcn with edge effective resistance. *Neurocomputing*, 423:389–398, 2021. 3

[2] Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021. 2

[3] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *ICCV*, pages 13339–13348, 2021. 1, 2, 3, 5, 6, 7

[4] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *ECCV*, page 536–553, Berlin, Heidelberg, 2020. Springer-Verlag. 3, 7

[5] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *CVPR*, pages 180–189, 2020. 3

[6] Hyung-Gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *CVPR*, pages 20154–20164, 2022. 1, 2, 3, 6, 7

[7] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, pages 3218–3226, 2015. 2

[8] Jeonghyeok Do and Munchurl Kim. Skateformer: skeletal-temporal transformer for human action recognition. In *ECCV*, pages 401–420. Springer, 2024. 3, 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015. 1

[11] Dong Feng, Zhongcheng Wu, Jun Zhang, and Tingting Ren. Multi-scale spatial temporal graph neural network for skeleton-based action recognition. *IEEE Access*, 9:58256–58265, 2021. 3, 7

[12] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn+: General hypergraph neural networks. *TPAMI*, 45(3):3181–3199, 2023. 2, 3

[13] Xiaoke Hao, Jie Li, Yingchun Guo, Tao Jiang, and Ming Yu. Hypergraph neural network for skeleton-based action recognition. *TIP*, 30:2263–2275, 2021. 2, 3, 7

[14] Ruijie Hou, Zhihao Wang, Ruimin Ren, Yang Cao, and Zhao Wang. Multi-channel network: Constructing efficient gcn baselines for skeleton-based action recognition. *Computers & Graphics*, 110:111–117, 2023. 3

[15] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Part-level graph convolutional network for skeleton-based action recognition. In *AAAI*, pages 11045–11052, 2020. 3

[16] Cloe Huesser, Simon Schubiger, and Arzu Çöltekin. Gesture interaction in virtual reality: A low-cost machine learning system and a qualitative assessment of effectiveness of selected gestures vs. gaze and controller interaction. In *International Conference on Human-Computer Interaction*, page 151–160, Berlin, Heidelberg, 2021. Springer-Verlag. 1

[17] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2012. 2

[18] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *IJCAI*, pages 2635–2641, 2019. 2

[19] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Multilevel hypergraph partitioning: Application in vlsi domain. In *Proceedings of the 34th annual Design Automation Conference*, pages 526–529, 1997. 2

[20] Matthew Korban and Xin Li. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *ECCV*, page 761–776, Berlin, Heidelberg, 2020. Springer-Verlag. 3

[21] Bokyung Lee, Michael Lee, Pan Zhang, Alexander Tessier, and Azam Khan. Semantic human activity annotation tool using skeletonized surveillance videos. In *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 312–315, 2019. 1

[22] Jungho Lee, Minhyeok Lee, Suhwan Cho, Sungmin Woo, Sungjun Jang, and Sangyoun Lee. Leveraging spatio-temporal dependency for skeleton-based action recognition. In *ICCV*, pages 10221–10230, 2023. 3

[23] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *ICCV*, pages 10410–10419, 2023. 2, 3, 5, 7

[24] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 3595–3603, 2019. 3

[25] Jun Liu, Amir Shahroudy, Dong Xu, Alex C. Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *TPAMI*, 40(12): 3007–3021, 2018. 1

[26] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *TPAMI*, 42(10):2684–2701, 2020. 1, 2, 6

[27] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, pages 140–149, 2020. 3, 7

[28] Ashish S. Nikam and Aarti G. Ambekar. Sign language recognition using image based hand gesture recognition techniques. In *2016 Online International Conference on Green Engineering and Technologies*, pages 1–5, 2016. 1

[29] Cosmas Ifeanyi Nwakanma, Fabliha Bushra Islam, Mareska Pratiwi Maharani, Dong-Seong Kim, and Jae-Min Lee. Iot-based vibration sensor data collection and emergency detection classification using long short term memory. In *International Conference on Artificial Intelligence in Information and Communication*, pages 273–278, 2021.

[30] Batyrkhan Omarov, Sergazy Narynov, Zhandos Zhumanov, Aidana Gumar, and Mariyam Khassanova. A skeleton-based approach for campus violence detection. *Computers, Materials & Continua*, 72(1), 2022. 1

[31] Yunsheng Pang, Qiuhong Ke, Hossein Rahmani, James Bailey, and Jun Liu. Igformer: Interaction graph transformer for skeleton-based human interaction recognition. In *European Conference on Computer Vision*, pages 605–622. Springer, 2022. 3

[32] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *Pattern recognition. ICPR international workshops and challenges: virtual event, January 10–15, 2021, Proceedings, Part III*, pages 694–701. Springer, 2021. 2, 3

[33] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv preprint arXiv:2201.02849*, 2022. 3

[34] Megani Rajendran, Chek Tien Tan, Indriyati Atmosukarto, Aik Beng Ng, and Simon See. Review on synergizing the metaverse and ai-driven synthetic data: enhancing virtual realms and activity recognition in computer vision. *Visual Intelligence*, 2(1):27, 2024. 1

[35] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016. 1, 2, 6

[36] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, pages 12018–12027, 2019. 2, 3, 5, 7

[37] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7904–7913, 2019. 3

[38] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *Proceedings of the Asian conference on computer vision*, 2020. 2, 3, 7

[39] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *TIP*, 29:9532–9545, 2020. 6

[40] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, page 568–576, Cambridge, MA, USA, 2014. MIT Press. 2

[41] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *TPAMI*, 45(2):1474–1488, 2022. 3, 7

[42] Ahmed Taha, Hala H Zayed, ME Khalifa, and El-Sayed M El-Horbaty. Skeleton-based human activity recognition for video surveillance. *International Journal of Scientific & Engineering Research*, 6(1):993–1004, 2015. 1

[43] Jianwei Tang, Jieming Wang, and Jian-Fang Hu. Predicting human poses via recurrent attention network. *Visual Intelligence*, 1(1):18, 2023. 1

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[45] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *ICCV*, pages 4041–4049, 2015. 1

[46] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *CVPR*, page 2649–2656, USA, 2014. IEEE Computer Society. 2, 6

[47] Qingtian Wang, Shuze Shi, Jiabin He, Jianlin Peng, Tingxi Liu, and Renliang Weng. Iip-transformer: Intra-inter-part transformer for skeleton-based action recognition. In *BigData*, pages 936–945, 2023. 2, 3, 7

[48] Shengqin Wang, Yongji Zhang, Hong Qi, Minghao Zhao, and Yu Jiang. Dynamic spatial-temporal hypergraph convolutional network for skeleton-based action recognition. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2147–2152. IEEE, 2023. 7

[49] Yizhe Wang, Congqi Cao, and Yanning Zhang. Visual-semantic network: a visual and semantic enhanced model for gesture recognition. *Visual Intelligence*, 1(1):25, 2023. 1

[50] Cong Wu, Xiao-Jun Wu, and Josef Kittler. Graph2net: Perceptually-enriched graph learning for skeleton-based action recognition. *TCSVT*, 32(4):2120–2132, 2022. 3

[51] Jianyang Xie, Yanda Meng, Yitian Zhao, Anh Nguyen, Xiaoyun Yang, and Yalin Zheng. Dynamic semantic-based spatial graph convolution network for skeleton-based human action recognition. *AAAI*, 38(6):6225–6233, 2024. 1, 3, 7

[52] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 2, 3, 5, 7

[53] Hengqi Zhang, Hua Geng, and Geng Yang. Two-stream transformer encoders for skeleton-based action recognition. In *International Conference on Computing, Control and Industrial Engineering*, pages 272–281. Springer, 2021. 2, 3

[54] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *CVPR*, pages 1109–1118, 2020. 3

[55] Yuhan Zhang, Bo Wu, Wen Li, Lixin Duan, and Chuang Gan. Stst: Spatial-temporal specialized transformer for skeleton-based action recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3229–3237, 2021. 3

[56] Huanyu Zhou, Qingjie Liu, and Yunhong Wang. Learning discriminative representations for skeleton based action recognition. In *CVPR*, pages 10608–10617, 2023. 7

[57] Yuxuan Zhou, Xudong Yan, Zhi-Qi Cheng, Yan Yan, Qi Dai, and Xian-Sheng Hua. Blockgcn: Redefine topology awareness for skeleton-based action recognition. In *CVPR*, pages 2049–2058, 2024. 2, 3, 7

[58] Yiran Zhu, Guangji Huang, Xing Xu, Yanli Ji, and Fumin Shen. Selective hypergraph convolutional networks for skeleton-based action recognition. In *ICMR*, page 518–526, New York, NY, USA, 2022. Association for Computing Machinery. 2, 3, 7