Skeleton-Based Action Recognition with Adaptive and Self-Attentive Graph Convolution Network

Ali Raza Shahid, Student Member, IEEE, Hong Yan, Fellow, IEEE

Abstract—Human skeleton-based action recognition in 3D sequences is an active research area in computer vision. The dynamics of human skeletons have received less attention in terms of semantics and joints connectivity. Graph Convolution Networks (GCNs) have recently shown promising performance for this task due to their strengths in modeling the dynamics and dependencies of sequential data. The human body structures’ spatial and temporal dynamics as a graph reflects the important information for the particular activity. However, existing GCN methods do not have the adaptive and self-attentive ability towards the adjacency matrix. This paper proposes a new class of GCN, adaptive local and global context-aware, and spatiotemporal self-attentive GCN, for skeleton-based action recognition. Adaptive and Self-Attentive graph convolution neural network (ASA-GCN) is capable of focusing on the important joints and bones in each frame by using local and global adaptive graph topology. To further improve the capability, we also introduced spatial and temporal self-attention graph mechanism, with which the attention performance of our network enhanced progressively. Extensive experiments on three large-scale datasets demonstrate that the proposed model performance exceeds the state-of-the-art with a significant margin.

Index Terms—Skeleton-based action recognition, Adaptive graph, Self-attention, Graph convolution network

I. INTRODUCTION

Human action recognition is a fast-growing research area due to its applications to human-computer interaction, intelligent robots, surveillance systems, and so on [1]. The earlier works [2] recognize the human actions from RGB video data, which involves complex backgrounds and varying illumination. With cost-effective depth devices [3], action recognition methods based on 3D human skeleton data attracted a lot of attention. Compared to the RGB videos, the skeleton-based methods are more robust and accurate [4], [5].

Recognizing an action from the 3D skeleton sequences, the joints spatial structure with bones and the temporal dynamics need to be exploited for a good understanding of the action classification [7]. Hidden Markov Models (HHMs) with multi-class AdaBoost [8] and Conditional Random Field (CRF) [9] performed a dynamic procedure to identify the activity using the motion sequence and parameter learning. Latent structural support vector machine (LSSVM) [6] and Naive Bayes Nearest-Neighbor (NBNN) classifier also presented to learn the temporal skeletal. These traditional models are based on handcrafted features that cannot effectively capture the long-term temporal structure and the discriminatory spatial information of the skeleton sequence [7]. However, the timestep of the action sequence video may contain ambiguous sub-actions and it is necessary to learn entire sequence precisely. The existing methods of skeleton-based action recognition encode all coordinates of the human body joints in every frame to a feature vector for pattern learning [8], [9], [10]. These models seldom analyze the internal dependencies that arise from lacking much action information between the joints and the bones.

Generalizing the recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to the structural graphs is an emerging topic in deep learning [11]. New generalizations and definitions are developed to handle the complexity of structured data, so a graph convolution is generalized from a 2D convolution and GNNs are evolved to overcome the CNNs complexity [12]. GCN performs generalized convolutions from image to graph [13], and has been adopted in many applications efficiently [14], [15], [16]. Natural structure of the human skeleton is the geometric space graph, where the joints as nodes and their natural bones relations in the human body as edges. Previous methods cannot exploit the graph structure of the skeleton data as it is difficult to generalize the skeleton in the spatiotemporal domain. The spatiotemporal GCN (ST-GCN) [17] is the first method to use a skeleton-based GCN model with the graph data to perform action recognition. The ST-GCN extracts the joints feature directly from a structure. The graphs are typically fixed over all layers, lacking the flexibility and ability to model multi-level semantics in various layers, which may not be suitable to various categories in human actions [18], [19]. For action categories such as "put on glasses" and "put on a hat/cap" hands and head relations should be greater, but this is not valid for some other classes, such as "clapping" and "rub two hands" because it depends on the movement and orientation of the hands. We propose the recognition mechanism that relies on the local and global structure functional knowledge with numerous relations that are not covered in the existing methods.

To solve the above problems, adaptive and self-attentive convolution block is presented in this paper. It defines the graph convolutions on three types of graphs. (1) In the first place, it is

This work is supported by the Hong Kong Innovation and Technology Commission and City University of Hong Kong (Project 9610034). (Corresponding author: Ali Raza Shahid)

A. R. Shahid, is with City University of Hong Kong, Kowloon 100071, Hong Kong (e-mail: a.raza@my.cityu.edu.hk).

H. Yan is with Department of Electrical Engineering, City University of Hong Kong, Kowloon 100071, Hong Kong (e-mail: hyan@cityu.edu.hk).
The local and global graph, which is obtained by learning the adjacency matrix based on the information acquired from the dataset and the relations established according to the region of attention. Thus, the obtained local and global graph topology is more suitable than the previous human body-based graph to recognize the action categories. (3) The graph of structure similarity, whose relations, according to the similarity between graph nodes, are built as edges. As the data samples vary, a unique structure for each input can be captured by the adaptive layer. The three types of graphs are fused, which can adapt to their importance in each layer block. This data-driven procedure increases the stability of the model with reduced parameters.

Furthermore, subsequently the mechanism of attention in many tasks has proven to be effective [5], [20], [21], [22], so for skeleton-based action recognition, it is important to investigate. For diverse action categories and data sequences, in spatial, temporal and channel dimension plays different roles. These findings led us to design a spatial and temporal layer to conform to joints and bones activation, followed by channel attention. Each adaptive graph convolution layer includes this self-attentive block. The efficiency of graph convolution is increased by adding a limited number of parameters.

Another important issue with the current methods is that the feature vector is obtained using 2D or 3D coordinate values. We refer it as first-order information of the skeleton data. However, the second-order information, such as bones between joints, is not explored. Moreover, since timesteps have shown a useful modality in time flow for skeleton-based action recognition [7], [23], we proposed to extract coordinate derivatives of joints between previous and subsequent frames as motion sequence information for modeling the time flow of action. Finally, joint information and bone information as streams together with the motion sequence stream, are merged into a multi-stream proposed framework.

To validate the effectiveness of the proposed method using ASA-GCN, experiments are performed using NTU-RGB-D 60 [24], NTU-RGB-D 120 [25] and Kinetics-Skeleton [26] datasets. Our proposed method achieved state-of-the-art performance. The main contributions of this work include the following:

**First**, an adaptive graph topology is proposed to learn the graph's topology in the end-to-end learning method adaptively. The sub-graph method for adaptive graph convolution better suits the action recognition because of a hierarchical structure of the GCN and diversified skeleton sequences as graphs.

**Second**, the self-attentive layer's block is proposed for spatial and temporal dimension together with channel attention and embed with graph convolution layer, which helps model to learn selectively focus area on discriminating the important feature in every dimension individually.

**Third**, the first order-joint stream information combined with second-order bone stream information brings notable improvements in recognizing action. Furthermore, we extracted the motion sequence information using derivatives of joints in temporal direction and integrated these three modalities in a multiple stream network.

## II. RELATED WORK

### A. Skeleton-Based Action Recognition

Skeleton-based action recognition has been exploited in many different ways in recent years [27], [28]. Conventional methods rely on handcrafted features to model the human body e.g., the super normal vector (SNV) [2], Lie Group [29]. The most widely used RNN and CNN methods usually model the spatiotemporal joints in skeleton data as vectors [4], [5], [24], [30]. Spatiotemporal long-term memory (ST-LSTM) [5] uses a gated model, encoding a skeleton by tree structure and contain them chain vector. A new LSTM network in [6] uses global context-aware attention (GCA-LSTM), which selectively focuses on the frame's informative joints using the attention mechanism. In [24], the authors introduced the two-layer part aware (2L-PA-LSTM) to model the temporal correlations of each body part. The view adaption recurrent neural network (VA-RNN) by [30] regulates the observation viewpoints on an action's occurrence.

CNN models the skeleton data as a spatial image based on designed transform rules [31], [7], [32], [33], [34]. In [31], a skeleton visualization two-stream CNN (SV-2s-CNN) method presents skeleton series as motion and visual color images in a compact form. The rotated skeleton in sequences mapped in clip and a multitask CNN (RC-MTCNN) to learn the generated clips for action recognition presented in [7]. In [32], the Euclidean distance matrix between global joints (JEDM) and local dynamic variation of the joint Euler angles (JEAs) is used as a feature. SkeletonNet [33] extract local features from frames of the skeleton sequence and instead of treating the features of all frames as a time series, the transformed features are feed CNN model. So, to achieve efficient sequence parametrization of RNNs and the better accuracy of CNNs, some hybrid methods are proposed [34], [35].

However, because the skeleton data is naturally embedded in graphs instead of a vector sequence or a 2D pseudo-grid, both the RNNs and CNNs cannot represent the skeleton data as a structure with low parameters.

### B. Graph Convolutional Neural Networks

The GCNs cannot be directly applied for many problems because of some of their baseline limitations [4]. In conventional neural networks, the image is taken as a convolution node to take the weighted average of pixels, and a neighbor node can be ordered and have fixed kernel [36]. So, the graph convolutions are extracted from conventional CNNs by representing irregular nodes [37], [38]. The operation is performed by taking the average node value and its unordered neighbors with different sizes, which is different from the image data.

The principle of constructing the GCNs on the graphs are generally divided into spectral and spatial perspective [59]. The spectral perspective, where the modality of the data type is spectral, and graph convolution is considered as the analysis of spectral information [37], [39], [40], [41]. However, the spectral GNNs input the graph structure after domain transformation, limiting the computational efficiency and generalizability to new graphs due to the assumption of fixed adjacency. In contrast, the spatial graph, where the convolution is applied directly on the spatial point and the neighboring
nodes, [42], [43], [44], [45]. However, spatial GNNs generally perform layer-wise operations for each node by selecting the neighbors and merging features with adjacent neighborhood nodes using aggregation and activation functions.

Using spatial GNNs, the body structure of the skeleton can naturally be embedded in the form of graphs as in [17], [19], [46], [18], [47]. The ST-GCN [17] modeled the skeleton data to achieve better accuracy. Based on that, [19] further proposed an adaptive graph learning method and improved, [46] used encoder-decoder structure to enrich actional structural (AS) graph convolution network. Semantic guided network (SGN) is presented to model the joint and frame-level modules by fusing the joint’s position and velocity as a dynamic representation [17]. In [47], a simple method to disentangle multi-scale (MS) graph convolutions using a unified spatiotemporal operator (G3D). Some methods use hybrid networks to improve spatiotemporal attention [22], [48]. An attention graph convolutions (AGC) capture a co-occurrence between spatial and temporal domains using hierarchical LSTM to increase temporal receptive fields [22]. In [48], deep progressing reinforcement learning (DPRL) is used to select the video clip’s quality frames of action.

Among all these approaches, researchers have emphasized the locality of the joints and position within the human body parts with the complexity of large training parameters. Empowering the adjacency during the training phase and the self-attention reduces the parameterization to overcome the weighting problem in long-range deep temporal learning. Thus, our proposed method aims to address this with adaptive and self-attentive graph learning to power the adjacency. In addition, to enlarge the receptive field when the network goes deeper, we used parallel hierarchical dilated convolutions to reduce the complexity of parameters efficiently.

III. GRAPH CONVOLUTIONAL NETWORKS

In this section, we describe the basic graphs in the context of graph convolution neural network and our proposed graph-based method for the human skeleton-based action recognition.

A. Graph Construction

The data from raw skeletons are represented in each frame by a vector sequence. The 3D coordinate values of the corresponding human joints are represented by each vector. Multiple frames in videos of different lengths for different samples are comprised to complete action sequence. In order to model the local and global structural information of human body joints and bones across both the spatial and temporal dimensions, we use a spatial-temporal graph. The spatial dimension refers to joints within the same frame, while the temporal dimension refers to all frames with the same joints. The example of the developed spatial and temporal skeleton graph is shown in Fig. 1a. The joints are depicted as the nodes with their natural bones in the human body as the spatial edges as described in Fig. 1a with blue lines. The corresponding joints in the previous and next frames are linked as seen in Fig. 1a for the temporal dimension. The coordinate location of each joint is determined as a node attribute. This is why the graph is intrinsic and is based on the natural configuration of the human body.

B. Convolutional Graphs

Using the graph definition above, we employed multiple layers of spatiotemporal graph convolution operations. A human skeleton graph is denoted as \( G = (N, E) \), where \( N = \{ n_1, \ldots, n_m \} \) is the set of \( M \) nodes representing joints and \( E \) is the edge set representing the bones captured by an adjacency matrix \( A \in \mathbb{R}^{M \times M} \) where initially \( A_{i,j} = 1 \), if an edge directs from \( n_i \) to \( n_j \), and \( A_{i,j} = 0 \) otherwise. The matrix \( A \) is symmetric for graph \( G \). Human action 3D sequences from video data as graph sequences have a node features set \( \mathcal{X} = \{ x_{t,n} \in \mathbb{R}^C \mid t \in \mathbb{Z}, F \geq t \geq 1, M \geq n \geq 1 \} \) represented as a feature tensor \( f \in \mathbb{R}^{C_{TXL+1}} \), where \( x_{t,n} = f_{t,n} \) presents the dimensional feature vector \( C \) for a node \( n \) at time \( t \) over a full-time frames \( F \). Hence, an input action is described by \( A \) structurally and by \( f \) feature element with \( f_i \in \mathbb{R}^{C_{TXL+1}} \) being the joint and bone features at time \( t \) of the network. \( W_l \in \mathbb{R}^{C_L \times C_L+1} \) represents the learnable weight matrix at layer \( L \) of the network. The human body nodes are used to perform the convolution operation on input data defined as \( f \) (feature) and \( A \) (graph structure). A layer-by-layer updating rule for convolutional graph can be applied to features at time \( t \) on network as:

\[
 f_{t+1}(n_i) = D^{-1/2} \tilde{A} D^{-1/2} f_t( \eta_j ) W_L( \eta_j ),
\]

where \( \tilde{A} = A^T + A \) is the self-loops with every node itself to keep identity features \( (A^T) \) and skeleton graph \( A, D \) is the diagonal degree matrix of an adjacency matrix \( A \). The term \( D^{-1/2} \tilde{A} D^{-1/2} f_t( \eta_j ) \), is represented to approximate the spatial mean features from the directed graph convolution.

C. Weighting Problem in Graphs

Under the spatial graph framework defined in (1), the existing method [46] utilizes adjacency matrix order polynomials to obtain structural knowledge scaled at the time \( t \) as follows:

\[
 f_{t+1}^t(n_i) = \sum_{k=0}^{K} \tilde{A} f_t( \eta_j ) W_L( \eta_j ),
\]

where \( K \) is scale control over graph structure, \( \tilde{A} \) is the normalized form of \( A \), for example [49] uses the symmetricity of graph regularization as \( \tilde{A} = A^T - D^{1/2} A D^{1/2} \) from directed graph convolution network. It can be seen that \( \tilde{A}_{i,j}^{k} = \tilde{A}_{i,j}^{k} \) corresponds to \( k \) length walks between \( n_i \) and \( n_j \), and thus the
term $\tilde{\mathbf{A}} f_i(n_j)$ is the weighted feature average based on the number of such random walks. This is not clear that how many others walk to nearer nodes than to real neighbors. This creates bias within the region as well as region with high degree of nodes. Hence, the GCN self-loop node makes for more feasible cycles and amplifies bias. To overcome such issues, we used the $k$ adjacency matrix, in which the $(i, j)$ entry is the shortest distance in a number of hops between $n_i$ and $n_j$. They defined cross-space-time skipping connection to distinguish each neighbor. $\tilde{\mathbf{A}}$ is computed using the differences in graph power as:

$$[f_x(n_i)]_{t+1} = \sum_{k=0}^{K} [\mathbf{D}^{-1/2} \tilde{\mathbf{A}} \mathbf{D}^{-1/2}] [f_x(n_i)]_t W_L(n_j),$$

where $\mathbf{D}^{-1/2} \tilde{\mathbf{A}} \mathbf{D}^{-1/2}$ is the disentangling and unifying adjacency matrix, $r$ is the number of dense connections of hops and itself nodes across all the frames. The connections with various nodes are represented by the spatial neighbors within the frames of single-action using the sliding window function, which makes the modeling graph complex and extensive connections. So, the input is dominated by amplifying dense graph connections between frames and then dilated windows to overcome the bias amplification using the large receptive fields. However, this makes the graph model ineffective in capturing the local parts information dependency for longer range and multiple attention regions.

**D. Local and Global Graph Implementation**

To overcome the identified problems, we define the graph convolution operation on global and local nodes formulated as:

$$f_{t+1}(n_i) = \sum_{n_j \in G_t} \tilde{\mathbf{A}}^{-1} f_i(n_j) W_L(m_i(n_j)), \quad (4)$$

here $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{L}^\delta$ is skeleton graph added with symmetric Laplacian graph matrix $\mathbf{L}^\delta = \mathbf{D}^{-1/2} (\mathbf{D} - \mathbf{A}) \mathbf{D}^{-1/2}$ keeps the features updated with the graph weights, $G_t$ represents the sampling region of the convolution for $n_i$, which is stated as the single neighboring distant node $n_j$ of the target node $n_i$. The weighting function $W_L$ is similar to the conventional convolution method, which provides a weight vector based on the input provided. The weight vector number is fixed, while the number of nodes in $G_t$ is fixed. The mapping function $m_i$ is therefore needed to map all local and global nodes into a fixed numbered sub-element, each associated with a particular weight vector. Fig. 1b shows mapping strategy, where the global graph is represented by $G_t$ and the local parts within the human body is represented by $G_{t1}$ upper right, $G_{t2}$ upper left, $G_{t4}$ lower right and $G_{t5}$ lower left sub-elements. As $G_{t1}$ represents the natural connections of nodes and edges so, it is added with every local sub-element to balance the contribution of each sub-element. The human skeleton graph and the related adjacency matrix for global structure and local sub-elements are shown in Fig. 2.

Specifically, the network’s feature map is basically a $C \times F \times M$ tensor, where the total number of joints is $M$ nodes, $F$ represents the temporal dimension and $C$ represents coordinates channel. For implementation, (4) is transformed into:

$$\mathbf{F}_{out} = \sum_{k} \mathbf{W}_k (\mathbf{F}_{in} \tilde{\mathbf{A}}_k) \odot (\mathbf{M}_k),$$

where $\mathbf{M}_k$ represents kernel size of the spatial dimension. Using local and global partition strategy design above, $\mathbf{M}_k$ is set to 5. $\tilde{\mathbf{A}}_k = \Lambda^{-1/2}_k \Lambda_k \Lambda_k^{-1/2}$, where $\Lambda_k$ is analogous to $M \times M$ adjacency matrix, and its element $\tilde{A}^{ij}_k$ denotes whether the node $n_j$ belongs to local graph $G_{k=1}$ of node $n_i$. For the corresponding weight vector, it is used to extract the connected nodes in a specific local element from $f_i$. $\Lambda^{-1/2} = A^{ij}_k + c$ is the diagonal matrix that is normalized. The weight vector $\mathbf{W}_k \in \mathbb{R}^{C \times F \times 1 \times 1}$ denotes 1X1 convolution operation to represent $W_i(m_i(n_j))$ as weighting function as in (4). $\mathbf{M}_k$ is an $M \times M$ attention mapping function that characterizes the importance of each node from the local and global graph, and $\odot$ represents the dot product operation.

In the case of the temporal dimension, for each node, the number of neighbours is fixed to two with corresponding joints in the previous and next frames. For initialization of temporal dimension computation, the sequence is padded. Similar to the classical convolutional operation, the graph convolution is now straightforward to perform. As shown in Fig. 3, $G_t$ is kernel size in temporal dimension and $G_{t} + 1$ convolution operation on the output feature maps are calculated.

With the above formulations, multiple layers of spatial and temporal graph convolution operations are performed. To classify and then predict the action classes based on the extracted attributes, the two fully connected layers with dropout are used.

**IV. ADAPTIVE AND SELF-ATTENTIVE GRAPH CONVOLUTION NETWORK**

This section introduces the proposed Adaptive and Self-Attentive Graph Convolutional Network (ASA-GCN) components in detail.

**A. Adaptive Graph Convolution Layer**

The spatial-temporal graph convolution for the skeleton data described above is calculated on the basis of an inherent human body structure-based graph, which may not be the best choice.
as explained in Section I. Adaptive and self-attentive graph convolution layers are proposed to solve this problem. It optimizes the topology of the graph along with other network parameters in an end-to-end way of learning. The graph is unique for different layers and samples, which greatly increases the model’s flexibility. In the meantime, it is designed as a residual branch that ensures the stability of the pipelined architecture.

According to details in (5), the graph is initially defined by the attention mask $M_k$ and the adjacency matrix $\tilde{A}_k$. The term $M_k$ determines the strength of connections in $\tilde{A}_k$, where the connections between the two nodes on the skeleton graph. To make the adaptive graph mechanism, we modify (5) as:

$$F_{out} = \sum_k G_k F_{in} (\tilde{A}_k + G_k + \lambda D_k),$$

(6)

The main difference is the adjacency matrix of the graph, which is divided into global and local elements. The first element $\tilde{A}_k$ is the same as original normalized $M \times M$ adjacency $\tilde{A}_k$ in (5). Specifically, this element in (6) represents the human body's physical structure as the original global graph.

The second element $G_k$ is also $M \times M$ adjacency matrix. In comparison to $\tilde{A}_k$, in the training process, the $G_k$ factor is parametrized and optimized along with other parameters. There are no limitations on the value of $G_k$, which means that the graph is fully learned by input training data ($x_i$) as in Fig. 3. The model learns graphs that are entirely targeted for action recognition in this data-driven manner and specify different information in different layers of local elements in the graph representation. Note that the element may be an arbitrary value in the matrix. It implies the significance of the relations between the two joints and the strength of the connections. As described in (5), this part can play the same function as the attention mask played by $M_k$. The previous attention map $M_k$ is dot multiplied to $\tilde{A}_k$, which ensures that if one of the values in element $\tilde{A}_k = 0$, it will still be 0 irrespective of the value of $M_k$, so that new connections that do not occur in the local graphs cannot be created. From this point of view, $G_k$ is more stable than $M_k$.

The third element of $D_k$ in (6) is data driven graph that learns new connections according to the input. For computing strength, we use the exponentials for the embedding parameters $p$ and $q$ with the multi-dimensional Gaussian function to calculate similarity between two nodes:

$$S(n_i, n_j) = \frac{\exp(p(n_i)^T q(n_j))}{\sum_j \exp(p(n_i)^T q(n_j))},$$

(7)

here in an embedding space the dot product of two nodes are measured. The input feature map $F_{in} \in \mathbb{R}^{C_v \times F \times M}$, is embed into the embedding space $\mathbb{R}^{C_v \times F \times M}$ with functions $p$ and $q$. By experiments, we select the 1X1 convolution layer as the embedding function. Embedded matrices of feature maps are reshaped in $M_{pk} \in \mathbb{R}^{C_v \times F \times M}$ and $M_{qk} \in \mathbb{R}^{F \times C_v \times M}$. A similarity matrix $D_k \in \mathbb{R}^{M \times M}$, is obtained by $M_{pk} \otimes M_{qk}$, and the similarity of the node $n_i$ and the node $n_j$ is denoted by $D_k$. The normalized matrix is used as edge between two nodes.

Although the simplified exponential is designed with a softmax layer ($\mathcal{L}$), based on (6), we can calculate $D_k$ as:

$$D_k = \mathcal{L} (F_{in} W_k^T W_q k F_{in}),$$

(8)

where $W_p \in \mathbb{R}^{C_v \times C_v \times F \times M}$ and $W_q \in \mathbb{R}^{C_v \times C_v \times F \times M}$ are the parameters of the embedding function $p$ and $q$ separately.

The global graph specifies the basic graph configuration for action classification, and the local graph adds uniqueness depending on the various discriminative characteristics of the sample. We found that the local graph is needed more strongly in the upper layers than in the lower layers. It is understandable in the architecture, as the bottom layer's receptive field is reduced, which limits the capability to learn graph features from various inputs. Apart from the lower layer, the information of the upper layer is more textual, and is more complex and requires more specific characteristics of the graph. The specific graph is easier to satisfy the requirement as it is based on input data and is unique to each of the input. Taking this into account, we use a weight multiplication ($\alpha$) parameter to adjust the value of each graph for different layers. Such a weight parameterization coefficient is different for each layer that is initialized, trained, and modified during the training process.

On the basis of the experiments, it is observed that in the early stage of the training process, the graph structure changes drastically, causing an unstable convergence of the model. To stabilize the training, we employed different strategies. The first is initializing the graph feature element matrix with $\tilde{A}_k + G_k + D_k$ as the adjacency, where $\tilde{A}_k$ is the fixed skeleton-based graph. The $G_k$ is incorporated using local body parts connections and $\alpha D_k$ is initialized with 1. The second strategy is initialization with $\alpha D_k$ with global graph $\tilde{A}_k$ only, thus $\tilde{A}_k$ dominates the early stage of the training. This approach shown to be better in experiments after initialization of the weights and the embedding functions. After $\tilde{A}_k + G_k + D_k$ operation, the collected $M \times M$ graphs are multiplied with Zeta ($\zeta$) parameter using the $w_i (1\times1)$ convolution layer. Besides this residual input of the graph $G_i F \times M$ is also multiplied with lambda ($\lambda$).
parameter using the \( w_2(1 \times 1) \) convolution layer. Basically, \( w_1 \) and \( w_2 \) separate parameters are then added together to prevent the early domination of \( \mathbf{A}_q \) graph during the training and blocking of gradient propagation at the initial phase of the training until the training stabilize.

The design of the adaptive graph convolution (AGC) layer is described in Fig. 3. The \( G_0 \) convolutional graph's kernel is equal to five. \( w_1 + w_2 \) are the weighting parameters presented in (5), results in \( \mathbf{W}_k \) as described in (6). A \( r(1 \times 1) \) residual branch is added along AGC layer to stabilize the phase of training and incorporation with attention layer. The resultant feature is represented by \( \mathbf{x}_0 \in \mathbb{R}^{C_0 \times F \times M} \).

![Fig. 4. Illustration of Attention Convolution layer. (1 x 1) denotes the size of the convolution kernel. The orange box denotes that the parameter is learnable. \( \oplus \) and \( \otimes \) represent the element-wise addition and multiplication respectively.](image)

**B. Spatiotemporal Self-Attention Layer**

The initial layers play a critical role in learning the features, which are later used to recognize the discriminative feature. There are several formulations for the attention mechanism constructed to preserve the feature form heavy correlated information. The existing attention mechanism focuses on the overall data in the neural networks and is used as a convolution layer for the whole perspective. A self-attentive layer is used to replace the convolution and build a full attention layer for data. With extensive experiments, we propose a self-attentive layer for 3D skeleton sequence action recognition which is described in Fig. 4. Furthermore, channel attention (CA) similar to [19], is also added after the self-attentive layers, allowing the attention mechanism to be completed over all the input data dimensions.

First, we describe a self-attentive layer to build full attention used to replace spatial convolutions. Similar to convolution, given input array \( \mathbf{x}_{ij} \in \mathbb{R}^{d_{in}} \), we first extract a local element for array in position \( ab \in N_c(ij) \) with element extent \( c \) centered around \( \mathbf{x}_{ij} \), which is known as a memory block. This form of local attention is different from the existing attention mechanism. Single-headed attention for computing the array element output \( \mathbf{y}_{ij} \in \mathbb{R}^{d_{out}} \) is then computed using the following:

\[
\mathbf{y}_{ij} = \sum_{a \in N_c(ij)} \mathcal{L}_{ab} \left( q_{ij}^T k_{ij} + q_{ij}^T c_1 c_2 \right) v_{ab}, \tag{9}
\]

where the queries \( q_{ij} = \mathbf{W}_q \mathbf{x}_{ab} \), key \( k_{ij} = \mathbf{W}_k \mathbf{x}_{ab} \), and value \( v_{ij} = \mathbf{W}_v \mathbf{x}_{ab} \) are linear transformations of the array element in position \( ij \) and the neighborhood array elements. \( \mathcal{L}_{ab} \) denotes a softmax applied to all logits computed in the neighborhood of \( ij \). \( \mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d_{in} \times d_{out}} \) are learned transforms. While, self-attentive aggregates element information over neighborhoods similar to convolutions, which is done with a combination of softmax parameterization by content interactions. This computation is repeated for every array element \( ij \). It works by partitioning the array features \( \mathbf{x}_{ij} \) depth-wise into \( C_0 \) groups \( \mathbf{x}_q \in \mathbb{R}^{d_{out}/C_0} \), computing single-headed attention on each group separately as above with different transforms \( \mathbf{W}_q^n, \mathbf{W}_k^n, \mathbf{W}_v^n \) per head, and then concatenating the output representations into the final output \( \mathbf{y}_{ij} \in \mathbb{R}^{d_{out}} \). As currently framed, no positional information is encoded in attention, which makes it permutation equivalent, limiting expressivity for recognition tasks. The earlier studies suggested that using relative positional embeddings results better accuracy. Instead, attention with relative position embeddings, a relative attention is used. Relative attention starts by defining the relative distance of \( ij \) to each position \( a b \in N_c(ij) \). The relative distance is factorized across dimension, so the element \( a b \in N_c(ij) \) receives distance of an array element offset \( a - i \) and array index offset \( b - j \). The array element and index are associated with embeddings \( c_1 \) and \( c_2 \) each with dimension \( 0.5xd_{out} \). These embeddings are concatenated after adding the additive learning parameters \( \alpha \) and \( \beta \) as shown in Fig. 4. This relation attention can be expressed as:

\[
\mathbf{y}_{ij} = \sum_{a \in N_c(ij)} \mathcal{L}_{ab} \left( q_{ij}^T k_{ij} + q_{ij}^T c_1 c_2 \right) v_{ab}, \tag{9}
\]

Thus, the logit measuring the similarity between the query and an element \( N_c(ij) \) is modulated both by the content of the element and relative distance of the element from the query. By using this method, the self-attentive layer also employs translation equivariance, similar to convolutions. Self-attentive layer is defined as attention applied to a single content instead of multiple contents using spatial self-attention (SSA) content and temporal self-attention (TSA) content.

The SSA layer will allow the model to concentrate on every joint at different level. So, using the operations described in (9) the \( \mathbf{y}_{SSA} \in \mathbb{R}^{1 \times 1 \times M} \) map computed as:

\[
\mathbf{y}_{SSA} = \phi \left( \mathbf{y}_{ij}(d_{in}) \right), \tag{10}
\]

where \( d_{in} \in \mathbb{R}^{C_0 \times F \times M} \) is the input feature map be an average of over all frames, \( \mathbf{y}_{ij} \) is one dimensional convolution operation. \( \phi \) is the ReLU activation function.

The TSA layer is similar to the \( \mathbf{y}_{SSA} \) and computed as:

\[
\mathbf{y}_{TSA} = \phi \left( \mathbf{y}_{ij}(d_{in}) \right), \tag{11}
\]
where \( y_{TSA} \in \mathbb{R}^{1 \times F \times T} \) and definitions of rest symbols are the same as (9).

The channel attention helps the architecture to strengthen the discriminative features according to input. The two fully connected layers are used to perform this task for channel \( y_{CA} \in \mathbb{R}^{C \times 1 \times T} \) which generates maps as follow:

\[
y_{CA} = \phi \left( F_{w1} \left( \phi \left( F_{w2}(d_{in}) \right) \right) \right),
\]

where \( F_{w1} \in \mathbb{R}^{C \times C} \) and \( F_{w2} \in \mathbb{R}^{C \times C} \) are the weight parameters for the fully connected layers. The attention layers introduced above can be placed differently, like parallel or series with a different order. After experiments, we found better results using the parallel of attention layers.

C. Basic Graph Convolution Block

The convolution performed on the input \( C_{in} \times F \times M \) feature map consists of three blocks: "ASA", "TCN" and "Residual" block, respectively. The architecture of the basic block is illustrated in Fig. 5. One basic "ASA Block" contains an AGC layer proceeded by batch normalization (BN), convolution and a rectified linear unit (ReLU) layer. Then \( y_{TSA}, y_{SSA} \) and \( y_{CA} \) are arranged to concatenate and capture the attention feature map. The residual connection is also added with attention layers introduced above can be placed differently, like parallel or series with a different order. After experiments, we found better results using the parallel of attention layers.

D. Network Architecture

The overview of the proposed ASA-GCN architecture is illustrated in Fig. 6 using the basic GCN blocks. There are five basic blocks included to train an activity recognition model \( (B_1 \text{ to } B_5) \). The number of input channels for each block is \( 3, 24, 48, 96, 144 \), and the output for the last channel 192. For the initialization of our model, a batch normalization layer is used to normalize the input data. After performing the block operations, the output is sent to a fully connected layer \( f c_1 \in \mathbb{R}^{192 \times 144} \). Dropout is used as a regularization technique to reduce the overfitting in our model and avoid complex skeleton data co-adaptations. After performing the experiments dropout of 0.25% is chosen to average the results effectively. Finally, the fully connected layer \( f c_2 \in \mathbb{R}^{144 \times N_C} \) gives the number of classification prediction \( N_C \).

E. Multi-Stream Fusion

Visualizing the human structural bones along with the joints information can help the system recognize the skeleton actions. This knowledge is also important for the recognition of skeletal activity but has been overlooked in past work. Beside joints and bones, a 3D sequence data information is also worth to be investigated for action recognition. Thus, a multi-stream system using a separate model of the same architecture is trained using
the joint, bone and sequence features, respectively. Specifically, each bone is connected with two joints, and we represent one joint as a source and another as a target. Each bone is interpreted as a vector by the distance data of the source joint \((a)\) and the destination joint \((b)\). For example, given a bone in frame \(\tau\) with its two joints \(m_a^\tau = (x_a^\tau, y_a^\tau, z_a^\tau)\) and \(m_b^\tau = (x_b^\tau, y_b^\tau, z_b^\tau)\), the vector of bone is calculated as:

\[
b_{m_a^\tau, m_b^\tau} = [(x_b^\tau - x_a^\tau), (y_b^\tau - y_a^\tau), (z_b^\tau - z_a^\tau)].
\]

where \(\tau\) is a number of frames \(F\), and \(a\) and \(b\) are the indices of source joint and target joint. Since the skeleton graphs are the tree-like structure, we added a zero-bone vector at the body to obtain \(M\) bones from the \(M\) joints and reuse the adjacency matrix for the connectivity in representation graphs. For the given samples, we first calculated the joints data for joint-stream. Then based on joints data, the bones data is calculated for the bone-stream.

As for sequence information, the derivative is calculated between the same joints in the next \((\tau + 1)\) and previous \((\tau - 1)\) frame with respect to the existing joints in the current frame \((\tau)\). For the first frame of joints, the previous frame is assumed zero. For example, the given joint in current frame \(\tau\) is \(m_a^\tau = (x_a^\tau, y_a^\tau, z_a^\tau)\) and the same joint in next \(m_{a+1}^\tau = (x_{a+1}^\tau, y_{a+1}^\tau, z_{a+1}^\tau)\) and previous \(m_{a-1}^\tau = (x_{a-1}^\tau, y_{a-1}^\tau, z_{a-1}^\tau)\) is represented as:

\[
m_a^\tau = m_{a-1}^\tau + m_{a+1}^\tau - 2m_a^\tau.
\]

where \(\tau\) is the same as given in (12). The streams of joints, bones, and their sequence derivatives are fed in into the network, and the action scores \(N_C\) of the last layer are summed to obtain the final prediction action label.

V. EXPERIMENTS

For comparison, the experiments are conducted on three action recognition datasets: NTU-RGB-D 60 [24], NTU-RGB-D 120 [53] and Kinetics-Skeleton 400 [26]. Using NTU-RGB-D 60, we perform extensive experiments because it’s smaller than NTU-RGB-D 120 and Kinetics-Skeleton 400. An ablation study is performed to verify the effectiveness of the proposed model components-based recognition performance. The final model then evaluated three datasets and validated the ability to generalize and compare them with other state-of-the-art approaches for skeleton-based action recognition tasks.

A. Datasets

1) NTU-RGB-D 60

It’s a large-scale and commonly used in-door 3D action recognition dataset. This dataset includes 56,880 video clips of 60 human action categories and 4.7 million frames. These clips are made by 40 volunteers aged 10 to 35 years old. Three cameras at the same height but with different horizontal angles capture each action \((+45^\circ, 0^\circ, -45^\circ)\). For every frame detected by the Kinect depth sensors, this dataset includes a 3D joint location. In the skeleton action sequences, there are 25 joints for each subject, but each clip has no more than two subjects. Two benchmarks are recommended in the original dataset paper: (1) Cross-subject (x-sub), the dataset in this benchmark is divided into a training set of 40,320 clips and a validation set of 16,560 clips, where the subjects are different in the two subsets. (2) Cross-view (x-view), the training set of this benchmark comprises 37,920 videos captured by camera 2 and 3 and 18,960 videos captured by camera 1 in the validation set.

2) NTU-RGB-D 120

The largest dataset of human action recognition annotations with 3D sequences. There are 114,480 action samples of 120 action classes and 8 million frames in the dataset. With three camera views, samples are captured among 106 volunteers. This data set contains 32 different camera configurations, each of which denotes a specific background and location. This dataset includes differences in environmental conditions, using 96 distinct backgrounds and including major variations in lighting. The actions are divided into three main groups: (a) 82 everyday actions (moving objects, sitting, eating, standing, writing, etc.), (b) 12 health-related actions (falling down, vomiting, blowing nose, etc.) and (c) 26 shared actions involving two people (hugging, handshaking, hitting, pushing, etc.).

The author of this dataset suggests two benchmarks: (1) cross-subject (x-sub) benchmark: the 103 subjects (ID’s from 1-103) are separated into 63,026 training samples, and the remaining three subjects (ID’s 104-106) reserved for testing 50,919 samples. (2) In benchmark for the cross-setup benchmark (x-set), 32 camera setups are used. The three camera setups are the same as the NTU-RGB-D 60 dataset in horizontal directions \((-45^\circ, 0^\circ, +45^\circ)\), and the rest are changes in the vertical height of the camera their distances to subjects. For experiments, the camera setups are divided equally for training and testing. There are 53 subjects used for training and the rest for testing. Moreover, the quantity is divided as training samples (54,468) using even ID numbers and testing samples (59,477) with an odd ID number.

3) Kinetics-Skeleton 400

Human actions that comprise 300,000 video samples for 400 action categories [26]. The video clips are taken from YouTube and include a wide variety of human actions. The Kinetics dataset includes raw video samples without details about the human skeleton. Using the freely accessible. The OpenPose toolbox for the human skeleton is derived from video clips [54]. Kinetics-Skeleton 400 comprises of 240,436 training samples and 19,796 sequences, each of which includes 18 body joints along coordinates values and confidence ratio. Skeleton frames with lower confidence are discarded. For tests, we use Kinetics-Skeleton 400 and report top-1 and top-5 accuracy for evaluation.

B. Training Details

Experiments are performed using the neural network architecture in Pytorch. The computations are performed on 2X NVIDIA GTX 1080 graphics cards, 64GB RAM and Intel i7-7800X CPU, 3.50GHz×12. Models are trained using a stochastic gradient with momentum = 0.9 as an optimization technique. The batch size is 16 for training and 32 for testing using cross-entropy as a loss function for back-propagate gradients. Weight decay is set at 0.0001. For Kinetics-Skeleton 400, a batch size of 128 is used for training and 256 for testing purposes.
For the NTU-RGB-D 60 dataset, each sample of the dataset contains a maximum of two samples. If the number of persons is less than 2, the second body is filled with 0. The maximum number of frames in one sequence is 300. Repeat samples with existing samples if the total number of frames is less than 300. At the 20th and 40th epoch's, the learning rate is set at 0.1 and divided by 10. At the 70th epoch, the training process is stopped. For NTU-RGB-D 120, the learning rate step is used at the 30th and 50th epochs. The inputs are pre-processed using the strategy used in [19] for normalization and translation. For the Kinetics-Skeleton 400 dataset, the learning rate is also set at 0.1 and divided by 10 at the 35th and 55th epochs. For reasonable performance comparison, no data augmentation is used in our study.

C. Ablation Study

This section examines the effectiveness of the modules proposed in ASA-GCN using the NTU-RGB-D 60 dataset.

1) Adaptive Convolution Block

As specified in section IV, our proposed AGC layer contains three types of sub-graphs. \( \mathbf{A}_k \) the original skeleton graph, \( \mathbf{G}_k \) the local and global graphs and \( \mathbf{D}_k \) the learnable graph. We tested the performance of each graph individually and their combinations. The results are presented as AGCL-A, AGCL-G, AGCL-D and combination in two pairs with one skipping in Table 1. It demonstrates that designed graphs bring significant improvements to the task of action recognition. With the addition of local graphics, the model achieves the best performance. Moreover, the individual and pairwise strategy's performance, we also tested the designed graph with all of the three combinations as AGCL in Table 1. It suggests that adding all the designs together is slightly better. Overall, the complete AGCL brings improvements of +4.6% and +2.1% on the x-sub and the x-view benchmarks, respectively, compared to the full-body graph (AGCL-A).

**TABLE I**

ACCURACIES ON THE NTU-RGB-D 60 DATASET. THE BODY BASED ORIGINAL GRAPH IS DENOTED BY \( A \) AS SHOWN IN (6), \( G \) DENOTES THE LEARNABLE GRAPH, AND \( D \) DENOTES THE LOCAL GRAPH DESCRIBED IN SECTION IV, RESPECTIVELY.

| Methods    | Params (M) | x-sub Accuracy | x-sub Accuracy |
|------------|------------|----------------|----------------|
| AGCL-A     | 0.39       | 82.7%          | 91.4%          |
| AGCL-G     | 0.39       | 86.1%          | 93.7%          |
| AGCL-D     | 0.39       | 82.5%          | 92.5%          |
| AGCL wo/A  | 0.63       | 86.5%          | 92.8%          |
| AGCL wo/G  | 0.63       | 83.2%          | 91.2%          |
| AGCL wo/D  | 0.63       | 85.4%          | 92.8%          |
| AGCL       | 0.75       | 87.3%          | 93.5%          |

**TABLE II**

ACCURACY ON NTU-RGB-D 60 DATASET FOR EACH OF THE ATTENTION LAYERS WITH VARIOUS ARRANGEMENT STRATEGIES. AGCL-A AND AGCL ARE THE SAME AS IN TABLE I, RESPECTIVELY. ADD MEANS ADDITION OF LAYERS AND STC MEAN CONCATENATION.

| Methods    | Params (M) | x-sub Accuracy | x-sub Accuracy |
|------------|------------|----------------|----------------|
| AGCL-A     | 0.39       | 82.7%          | 91.4%          |
| STSAGC-S   | 0.60       | 83.1%          | 92.7%          |
| STSAGC-T   | 0.60       | 83.5%          | 93.1%          |
| STSAGC-C   | 0.46       | 81.0%          | 91.2%          |
| STSAGC-ADD | 0.89       | 82.9%          | 92.9%          |
| STSAGC-STC | 0.89       | 85.9%          | 93.4%          |
| AGCL       | 0.75       | 87.3%          | 93.5%          |
| ASA-GCN-ADD| 1.24       | 87.1%          | 94.1%          |
| ASA-GCN-STC| 1.24       | 88.4%          | 94.5%          |

2) Self-Attention Block

The efficacy of the proposed spatiotemporal self-attention graph convolution mechanism introduced in Section IV is verified in this section. Table II shows the results. The contributions of three attention layers are first tested separately, i.e. SSA, TSA, CA based on Self-Attention, shown as STSAGC-S, STSAGC-T and STSAGC-C, respectively. All three self-attention layers lead to performance enhancements. We then evaluate the effectiveness of adding and concatenating each of the attention layers in parallel, shown as STCAGC-ADD and STCAGC-STC, respectively. It is suggested that concatenating the spatial, temporal, and channel attention layers is better. Thus, we embed the STC self-attention block into AGCL and obtain results. It can be seen that the self-attention module improvement for AGCL (+1.1% and +1.0%) is less significant than that for AGCL-A (+5.7% and +3.1%). We conclude that the AGCL is better and the accuracy is already significant, so the impact of self-attention is slight. The learned training and tested features are visualized in Fig. 10 to see the classes in the hyperplane in last block of proposed architecture.

**TABLE III**

ACCURACY USING DIFFERENT INPUT MODALITIES ON THE NTU-RGB-D 60 DATASET. JB-ASA-GCN DENOTES THE JOINT AND BONES TOGETHER MS-ASA-GCN REPRESENTS ALL THE MODALITIES (JOINT, BONE, SEQUENCE). ASA-GCN REPRESENTS ASA-GCN-STC IN TABLE II.

| Methods    | Params (M) | x-sub Accuracy | x-sub Accuracy |
|------------|------------|----------------|----------------|
| J-ASA-GCN  | 1.24       | 88.4%          | 94.5%          |
| B-ASA-GCN  | 1.24       | 86.3%          | 91.4%          |
| S-ASA-GCN  | 1.24       | 86.0%          | 92.7%          |
| JS-ASA-GCN | 2.48       | 89.1%          | 93.0%          |
| BS-ASA-GCN | 2.48       | 88.5%          | 93.2%          |
| JB-ASA-GCN | 2.48       | 89.1%          | 95.4%          |
| MS-ASA-GCN | 4.96       | 90.4%          | 95.6%          |

![Fig. 7. The (a) training and (b) testing feature visualization of ASA-GCN-STC model using the NTU-RGB-D 60 dataset.](image-url)
accuracy is improved as expected. The performance of the sequence modality using the derivatives to capture the motion of the human body with single streams is generally lower than the joint and bone modality JS-ASA-GCN and BS-ASA-GCN, respectively. However, adding a single stream together still brings improvement in comparison to MS-ASA-GCN.

D. Comparison with the State-of-the-art Methods

We compared the final model with existing human action recognition methods on the NTU-RGB-D 60, 120 and Kinetics-Skeleton 400. Table IV, V, and VI show the results. The state-of-the-art methods used for the comparisons include the handcrafted feature-based methods, RNN/LSTM feature-based methods, deep CNN methods, and GCN methods. Our model achieves state-of-the-art results, which demonstrates the effectiveness of our model using the three datasets.

1) Results and Analysis of NTU-RGB-D 60

As in the 2L-PA-LSTM [24], the evaluation is performed on this dataset using two standard benchmarks, i.e., cross-subject (x-sub) and cross-view (x-view) evaluation. The results are shown in Table IV. The multiple streams (MS) adaptive and self-attentive graph convolution network (ASA-GCN) outperforms the other methods in both x-sub and x-view benchmarks. The accuracy of the proposed MS-ASA-GCN is 90.4% when tested with the x-sub benchmark dataset. In comparison to existing method MS-G3D Net-J [47], the accuracy is enhanced by +1%. When tested with the x-view benchmark, the accuracy is improved from 95.1% (2s-AGCN [19]) to 95.6%. Fig. 8 shows the comparison between network parameters versus accuracy for the skeleton based activity recognition methods. It can be seen clearly that our proposed method outperforms the state-of-the-art methods in terms of network parameters (J-ASA-GCN) and accuracy (MS-ASA-GCN). For the GCN-based methods, the number of parameters is very low compared to RNN/LSTM and CNN methods.

The improvement in the performance of the proposed method is due to the introduced multiple streams. As shown in Table IV, the GCN methods achieved higher accuracy than the non-graph methods in many cases. The RNN/LSTM networks perform better than the handcrafted methods as AGC-LSTM [22] achieved 89.2% and 95.0% for both protocols because RNN/LSTM methods are easy to capture the temporal information. Still, some LSTM methods (i.e., ST-LSTM [4]), have difficulty capturing temporal information with many timesteps. It can also be noted the hybrid methods like VARNN+CNN [34] got the highest accuracy in comparison to the handcrafted and RNN/LSTM methods. As opposed to LSTM, CNN is also capable of capturing the long-term temporal sequences of human skeleton, but still does not achieve the higher accuracy comparable to GCN. The two-stream CNN method 2s-JEDM+JEA’s [32] obtained improved accuracy (86.7% and 91.8%), which is higher than the hybrid (CNN+GCN) method DPRL GNN [48] and GCN method ST-GCN [17]. It can be observed from Table IV that the performance of the GCN based method is higher than CNN, RNN/LSTM, and handcrafted methods. Our GCN based method further improved the accuracy from the baseline GCN method ST-GCN [17] (81.5% and 88.3%) to +8.9% and 7.3% in both benchmarks (90.4% and 95.6%).

Our proposed method’s confusion matrix is tested with the NTU-RGB-D 60 dataset with x-view benchmark shown in Fig. 9a. It can be seen that some actions are quite similar to each other so, they are misclassified with each other. For example, “12-writing” and “11-reading”, “11-reading” and “29-playing with phone/tablet”, “53-pat on back” and “52-pushing”, etc. The classes “15-take of jacket”, “put on a shoe” and “put on hat/cap” are the top accurate actions and “11-reading”, “54-point finger” are the top misclassified actions. Using the skeleton sequence information without texture information makes the action recognition more challenging to some classes in which objects are involved, e.g., phone, pen, book, shoe, jacket, etc.

TABLE IV

| Features   | Methods          | Year | x-sub Accuracy | x-view Accuracy |
|------------|-------------------|------|----------------|-----------------|
| Handcrafted| SNV [2]           | 2017 | 59.8%          | 63.6%           |
|            | Lie Group [29]    | 2016 | 61.4%          | 67.0%           |
| RNN/LSTM   | 2L-PA-LSTM [24]   | 2016 | 62.9%          | 70.3%           |
|            | VA-RNN [30]       | 2017 | 79.4%          | 87.6%           |
|            | ST-LSTM [4]       | 2018 | 69.2%          | 77.7%           |
|            | GCA-LSTM [51]     | 2018 | 77.1%          | 85.0%           |
|            | CNN+LSTM          | 2018 | 82.4%          | 89.1%           |
|            | GLAN+SSAN [35]    | 2019 | 88.7%          | 94.3%           |
|            | VA-RNN+CNN [34]   | 2019 | 88.7%          | 94.3%           |
| CNN        | SkeletonNet [33]  | 2017 | 75.9%          | 81.2%           |
|            | SV-2s-CNN [31]    | 2017 | 80.0%          | 87.2%           |
|            | RC-MTCNN [7]      | 2018 | 81.1%          | 87.4%           |
|            | 2s-JEDM+JEA’s [32]| 2019 | 86.7%          | 91.8%           |
|            | CNN+GNN           | 2018 | 83.5%          | 89.8%           |
| LSTM+GNN   | DPRL GNN [48]     | 2018 | 83.5%          | 89.8%           |
|            | AGC-LSTM [22]     | 2019 | 89.2%          | 95.0%           |
| GCN        | ST-GCN [17]       | 2018 | 81.5%          | 88.3%           |
|            | AS-GCN [46]       | 2019 | 86.8%          | 94.2%           |
|            | 2s-AGCN [19]      | 2019 | 88.5%          | 95.1%           |
|            | SGN [18]          | 2020 | 89.0%          | 94.5%           |
|            | MS-G3D Net-J [47] | 2020 | 89.4%          | 95.0%           |
|            | MS-ASA-GCN (ours) | 2021 | 90.4%          | 95.6%           |

2) Results and Analysis of NTU-RGB-D 120

This large-scale dataset is captured from various horizontal and vertical viewpoints. As in APSR [25], the evaluation on this dataset is performed with two benchmarks, i.e., cross-subject (x-sub) and cross-setup (x-set). We follow the same benchmark to validate accuracy of our method. Table V shows the results. Our method performs better than the existing LSTM and CNN. When tested on the x-sub and x-set benchmarks the proposed MS-ASA-GCN achieves an accuracy of 87.1% and 89.8%, which is +23.1% and +23.7% better than APSR [25] in both
benchmarks. The APSR [25] achieved the highest accuracy of 64.0% and 66.1% among the LSTM based methods. The improvements are due to the local and global graph adaptive learning method. Table V shows that the average performance using the x-sub benchmark for MS-ASA-GCN is 87.1%, which is quite similar to MS-G3D Net-J [47] (86.9%). The improvement of our proposed method compared to the 2s-AGCN [19] method uses a self-attention mechanism other than the adaptive graph learning averaged over +4.2% and +4.9% on both benchmarks. It can be observed from Table V the GCN-based methods learn the spatial and temporal relations better than LSTM and CNN.

The confusion matrix of the proposed method for the NTU-RGB-D 120 dataset using the x-set benchmark is shown in Fig. 9b. We then performed action-wise analysis for the top accurate and top misclassified action classes, as shown in Fig. 9b. The classes “12-writing” and “73-staple book” are highly misclassified, and "27-jump up” and "98-arm swings” are top accurate actions. It has been observed that actions with motion obtained higher accuracy as compare to actions involving any object. The higher accuracy is obtained because of adaptive graph connections during the training process together with the attention mechanism.

Fig. 9. Confusion matrices of J-ASA-GCN on (a) NTU-RGB-D 60 (x-view) and (b) NTU-RGB-D 120 (x-set). The numbers on the vertical side of the axis correspond to the actual number of classes, and the horizontal axis shows the predicted classes.

**TABLE V**

| Feature | Methods | NTU-RGB-D 120 |
|---------|---------|---------------|
|         | Year    | x-sub Accuracy | x-set Accuracy |
| LSTM    | 2L-PA-LSTM [24] | 2016 | 25.5% | 26.3% |
| ST-LSTM [4] | 2018 | 55.7% | 57.9% |
| GCA-LSTM [5] | 2018 | 58.3% | 59.2% |
| APSR [25] | 2020 | 64.0% | 66.1% |
| CNN     | SV-1s-CNN [31] | 2017 | 60.2% | 63.3% |
| RC-MT-CNN [7] | 2018 | 62.2% | 61.8% |
| FS Net [1] | 2020 | 59.9% | 62.4% |
| GCN     | 2s-AGCN [19] | 2019 | 82.9% | 84.9% |
| SGN [18] | 2020 | 79.2% | 81.5% |
| MS-G3D Net-J [47] | 2020 | 86.9% | 88.4% |
| MS-ASA-GCN (ours) | 2021 | 87.1% | 89.8% |

**VI. CONCLUSION AND FUTURE WORK**

In this paper, we presented a new approach for modeling human activity recognition. We have proposed the adaptive and self-attentive graph convolution network (ASA-GCN) for human skeleton action recognition. The convolutional graph networks are more efficient than traditional convolutional neural networks in terms of network parameters. It also confirms that adaptively trained local and global graph interactions are more appropriate for the classification of human actions than natural human body structure-based graphs. The self-attentive layer mechanism is embedded in each graph convolution layer, which lets the model pay more attention to important interactions over spatial, temporal and channel dimensions. In addition, the concatenation of different scaled dilated convolution layers over the temporal domains brings additional improvements with a reduced number of network parameters. The finalized model is evaluated on two laboratory-controlled and one wild large-scaled datasets, i.e., NTU-RGB-D 60, 120 and Kinetics-Skeleton 400. Our method achieves state-of-the-art performance on three datasets.

Future work can focus on the combination of both skeleton data and human pose estimation from the video streams in a unified framework. Efficiently handling the corrupted skeleton joints from the video sequence in an end-to-end manner is also a challenging research gap that needs to be addressed in the future, observed while performing the experiments Kinetics-Skeleton 400 dataset.

**ACKNOWLEDGEMENT**

Authors would like to thank ROSE Lab at Nanyang Technology University, Singapore for NTU-RGB-D dataset release. A. R. Shahid would like to thank Prof. Q. Chen from Hong Kong University of Science and Technology.

3) Results and Analysis of Kinetics-Skeleton 400

As in ST-GCN [17], we follow the same evaluation protocol to compute the top-1 and top-5 accuracies. The comparison of the proposed method is shown in Table VI. It can be seen that the accuracy of the MS-ASA-GCN (38.7% and 60.9%) is quite similar with 2s-AGCN [19] (36.1% and 58.7%) and MS-G3D Net [47] (38.0 and 60.9) methods in both top-1 and top-5 accuracies, respectively. Our MS-ASA-GCN graph-based method can learn the spatial and temporal relations from the wild data as compared to the 2L-PA-LSTM [24] and 13D-TCN [55] based methods. Unlike 2s-AGCN [19] and MS-G3D Net [47], our proposed method does not use pre-processing to handle the noisy joints and bones in skeleton data. MS-ASA-GCN still performs better because of addition sequence derivatives. This is clearly shown that the features learned from the multiple streams are robust to noise due to the joints, bones and sequence modalities. Most of the actions are misclassified because of insufficient posed data, which is challenging to obtain accurate results among the 400 classes.

**TABLE VI**

| Feature | Methods | Kinetics-Skeleton 400 |
|---------|---------|-----------------------|
|         | Year    | Top-1 Accuracy | Top-5 Accuracy |
| LSTM    | 2L-PA-LSTM [24] | 2016 | 16.4% | 35.3% |
| CNN     | 13D-TCN [55] | 2017 | 20.3% | 40.0% |
| GCN     | ST-GCN [17] | 2018 | 30.7% | 52.8% |
|         | 2L-PA-LSTM [47] | 2019 | 36.1% | 58.7% |
|         | MS-G3D Net [47] | 2020 | 38.0% | 60.9% |
|         | MS-ASA-GCN (ours) | 2021 | 38.7% | 61.1% |

In addition, the concatenation of different scaled dilated convolution layers over the temporal domains brings additional improvements with a reduced number of network parameters. The finalized model is evaluated on two laboratory-controlled and one wild large-scaled datasets, i.e., NTU-RGB-D 60, 120 and Kinetics-Skeleton 400. Our method achieves state-of-the-art performance on three datasets.

Future work can focus on the combination of both skeleton data and human pose estimation from the video streams in a unified framework. Efficiently handling the corrupted skeleton joints from the video sequence in an end-to-end manner is also a challenging research gap that needs to be addressed in the future, observed while performing the experiments Kinetics-Skeleton 400 dataset.

**ACKNOWLEDGEMENT**

Authors would like to thank ROSE Lab at Nanyang Technology University, Singapore for NTU-RGB-D dataset release. A. R. Shahid would like to thank Prof. Q. Chen from Hong Kong University of Science and Technology.
REFERENCES

[1] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, “Skeleton-Based Online Action Prediction Using Scale Selection Network,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 6, pp. 1453–1467, Jun. 2020.

[2] X. Yang and Y. L. Tian, “Super Normal Vector for Human Activity Recognition with Depth Cameras,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 5, pp. 1028–1039, May 2017.

[3] Z. Zhang, “Microsoft kinect sensor and its effect,” IEEE Multimedia, 2012.

[4] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, “Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 3007–3021, 2018.

[5] J. Liu, G. Wang, L. Y. Duan, K. Abdiyeva, and A. C. Kot, “Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks,” IEEE Trans. Image Process., vol. 27, no. 4, pp. 1586–1599, 2018.

[6] X. Wu, D. Xu, L. Duan, J. Luo, and Y. Jia, “Action recognition using multilevel features and latent structural SVM,” IEEE Trans. Circuits Syst. Video Technol., 2013.

[7] Q. Ke, M. Bennisirou, S. An, F. Sohel, and F. Boussaid, “Learning Clip Representations for Skeleton-Based 3D Action Recognition,” IEEE Trans. Image Process., vol. 27, no. 6, pp. 2842–2855, 2018.

[8] Z. Hussain, Q. Z. Sheng, and W. E. Zhang, “A review and categorization of techniques on device-free human activity recognition,” Journal of Network and Computer Applications, 2020.

[9] Q. Li, R. Gravina, Y. Li, S. H. Alsamhi, F. Sun, and G. Fortino, “Multi-user activity recognition: Challenges and opportunities,” Inf. Fusion, vol. 63, pp. 121–135, 2020.

[10] M. Dimiccoli, A. Cartas, and P. Radeva, “Chapter 6 - Activity recognition from visual lifelogs: State of the art and future challenges,” in Multimodal Behavior Analysis in the Wild, X. Alameda-Pineda, E. Ricci, and N. Sebe, Eds. Academic Press, 2019, pp. 121–134.

[11] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” IEEE Trans. Neural Networks, vol. 20, no. 1, pp. 61–80, 2009.

[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” IEEE Trans. Neural Networks Learn. Syst., vol. XX, no. Xx, pp. 1–21, 2020.

[13] M. Edwards and X. Xie, “Graph based convolutional neural network,” in British Machine Vision Conference 2016, BMVC, 2016, 2016.

[14] D. Duvenaud et al., “Convolutional networks on graphs for learning molecular fingerprints,” in Advances in Neural Information Processing Systems, 2015.

[15] T. Kipf, E. Fetaya, K. C. Wang, M. Welling, and R. Zemel, “Neural relational inference for Interacting systems,” in 35th International Conference on Machine Learning, ICML 2018, 2018.

[16] M. Gu, Z. Zhao, W. Jin, R. Hong, and F. Wu, “Graph-based Multi-interaction Network for Video Question Answering,” IEEE Trans. Image Process., p. 1, 2021.

[17] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 2018, pp. 7444–7452.

[18] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, “Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1109–1118.

[19] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 12018–12027, 2019.

[20] F. Wang et al., “Residual attention network for image classification,” in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017.

[21] P. Ramachandran et al., “Stand-alone self-attention in vision models,” Adv. Neural Inf. Process. Syst., vol. 32, 2019.

[22] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, “An attention enhanced graph convolutional lstm network for skeleton-based action recognition,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019.

[23] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Advances in Neural Information Processing Systems, 2014.

[24] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, “NTU RGB+D: A large scale dataset for 3D human activity analysis,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 1010–1019, 2016.

[25] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Duan, and A. C. Kot, “NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 10, pp. 2684–2701, Oct. 2020.

[26] J. Carreira and A. Zisserman, “Quo Vadis, action recognition? A new model and the kinetics dataset,” in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017.

[27] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Learning actionlet ensemble for 3D human action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., 2014.

[28] A. Shahroudy, T. T. Ng, Q. Yang, and G. Wang, “Multimodal Multipart Learning for Action Recognition in Depth Videos,” IEEE Trans. Pattern Anal. Mach. Intell., 2016.

[29] R. Vemulasapalli and R. Chellappa, “Rolling rotations for recognizing human actions from 3D skeletal data,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[30] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, “View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition from Skeleton Data,” in Proceedings of the IEEE International Conference on Computer Vision, 2017.
[31] M. Liu, H. Liu, and C. Chen, “Enhanced skeleton visualization for view invariant human action recognition,” Pattern Recognit., 2017.

[32] Q. Nie, J. Wang, X. Wang, and Y. Liu, “View-Invariant Human Action Recognition Based on a 3D Bio-Constrained Skeleton Model,” IEEE Trans. Image Process., vol. 28, no. 8, pp. 3959–3972, 2019.

[33] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, “SkeletonNet: Mining Deep Part Features for 3-D Action Recognition,” IEEE Signal Process. Lett., vol. 24, no. 6, pp. 731–735, Jun. 2017.

[34] P. Zhang et al., “View Adaptive Neural Networks for High Performance Skeleton-Based Human Action Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1963–1978, 2019.

[35] Z. Yang, Y. Li, J. Yang, and J. Luo, “Action Recognition with Spatio-Temporal Visual Attention on Skeleton Image Sequences,” IEEE Trans. Circuits Syst. Video Technol., 2018.

[36] H. Gao and S. Ji, “Graph U-nets,” in 36th International Conference on Machine Learning, ICML 2019, 2019.

[37] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 2017.

[38] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in 33rd International Conference on Machine Learning, ICML 2016, 2016.

[39] X. Hao, J. Li, Y. Guo, T. Jiang, and M. Yu, “Hypergraph Neural Network for Skeleton-based Action Recognition,” IEEE Trans. Image Process., p. 1, 2021.

[40] N. Leonardi and D. Van De Ville, “Tight wavelet frames on multislice graphs,” IEEE Trans. Signal Process., 2013.

[41] R. Li, S. Wang, F. Zhu, and J. Huang, “Adaptive graph convolutional neural networks,” 32nd AAAI Conf. Artif. Intell. AAAI 2018, pp. 3546–3553, 2018.

[42] S. Abu-El-Haija et al., “Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing,” in 36th International Conference on Machine Learning, ICML 2019, 2019.

[43] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in Advances in Neural Information Processing Systems, 2017.

[44] K. Xu, S. Jegelka, W. Hu, and J. Leskovec, “How powerful are graph neural networks?,” in 7th International Conference on Learning Representations, ICLR 2019, 2019.

[45] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., 2013.

[46] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 3590–3598, 2019.

[47] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, “Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 140–149.

[48] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, “Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[49] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, “Spatio-temporal graph convolution for skeleton based action recognition,” in 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 2018.

[50] M. Längkvist, L. Karlsson, and A. Loutfi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” Pattern Recognit. Lett., 2014.

[51] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, 2016.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[53] J. Liu, A. Shahrourdy, M. Perez, G. Wang, L. Y. Duan, and A. C. Kot, “NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 10, pp. 2684–2701, Oct. 2020.

[54] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017.

[55] T. S. Kim and A. Reiter, “Interpretable 3D Human Action Analysis with Temporal Convolutional Networks,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2017.

Ali Raza Shahid received an MSc degree in electrical engineering from COMSATS University Islamabad, Pakistan, in 2015. He is currently pursuing the Ph.D. degree in Electrical Engineering at City University of Hong Kong. His research interests include pattern recognition and medical imaging.

Hong Yan received his Ph.D. degree from Yale University. He was Professor of Imaging Science at the University of Sydney and is currently Chair Professor of Computer Engineering and Wong Chung Hong Professor of Data Engineering at City University of Hong Kong. His research interests include image processing, pattern recognition, and bioinformatics. Professor Yan is an IEEE Fellow and IAPR Fellow, and he received the 2016 Norbert Wiener Award from the IEEE SMC Society for contributions to image and biomolecular pattern recognition techniques. He is a member of the European Academy of Sciences and Arts.