Computers, Materials & Continua
DOI:10.32604/cmc.2022.022783
Article

Tech Science Press

Skeleton Split Strategies for Spatial Temporal Graph Convolution Networks

Motasem S. Alsawadi* **and Miguel Rio**

Electronic and Electrical Engineering Department, University College London, London, WC1E 7JE, England
*Corresponding Author: Motasem S. Alsawadi. Email: motasem.alsawadi.18@ucl.ac.uk

Received: 18 August 2021; Accepted: 24 September 2021

Abstract: Action recognition has been recognized as an activity in which individuals' behaviour can be observed. Assembling profiles of regular activities such as activities of daily living can support identifying trends in the data during critical events. A skeleton representation of the human body has been proven to be effective for this task. The skeletons are presented in graphs form-like. However, the topology of a graph is not structured like Euclidean-based data. Therefore, a new set of methods to perform the convolution operation upon the skeleton graph is proposed. Our proposal is based on the Spatial Temporal-Graph Convolutional Network (ST-GCN) framework. In this study, we proposed an improved set of label mapping methods for the ST-GCN framework. We introduce three split techniques (full distance split, connection split, and index split) as an alternative approach for the convolution operation. The experiments presented in this study have been trained using two benchmark datasets: NTU-RGB + D and Kinetics to evaluate the performance. Our results indicate that our split techniques outperform the previous partition strategies and are more stable during training without using the edge importance weighting additional training parameter. Therefore, our proposal can provide a more realistic solution for real-time applications centred on daily living recognition systems activities for indoor environments.

Keywords: Skeleton split strategies; spatial temporal graph convolutional neural networks; skeleton joints; action recognition

1 Introduction

Action recognition (AR) has been recognized as an activity in which individuals' behaviour can be observed. Assembling profiles of regular activities such as activities of daily living (ADL) can support identifying trends in the data during critical events. These include actions that might compromise a person's life. For that reason, human AR has become an active research area. Generally, human activity is characterized by different recipes. Amidst these recipes, wearable sensor-based recognition systems have become one of the most utilized approaches. In this kind of system, the input data comes from a sensor or a network of sensors [1]. These sensors are worn by the person performing the action. In general, a sensor-based recognition system consists of a set of sensors and a central node [2]. The aim of this node is to compute the action representation and perform the action recognition. However, these sensor devices are seldom ergonomic. Hence, the discomfort and need of wearing an external device on a daily basis prevails. These characteristics cause that the person who is monitored to usually forgets to use the sensor device which makes the recognition system unfunctional [3].

Other AR solutions include computer vision-based systems such as optical flows, appearance, and body skeletons [4–6]. The use of dynamic human skeletons (DHS) usually carry vital information that encompasses other modalities. One of the main benefits of this approach is that it minimizes the need for wearing sensors. Therefore, to collect the data, surveillance cameras can be mounted on the ceiling or walls of the environment of interest; ensuring an efficient indoor monitoring system [7]. However, DHS modelling has not yet been fully explored.

A performed action is typically described by a time series of the 2D or 3D coordinates of human joint positions [6,8]. Furthermore, action is recognized by examining the motion patterns. A skeleton representation of the human body has been proven to be effective for this task. It provides a robust solution to noise, and it is considered to be a computational and storage-efficient solution [8]. Additionally, it provides a background-free data representation to the classification algorithms. This allows the algorithms to focus only on the human body pattern recognition without being concerned about the surrounding environment of the performed action scenarios. This work aims to develop a unique and efficient approach for modelling the DHS for human AR.

1.1 Open Pose

There are multiple sources of camera-based skeleton data. Recently, Cao et al. [9] released the open-source library *OpenPose* which allows real-time skeleton-based human detection. Their algorithm outputs the skeleton graph represented as an array with the 2D and the 3D coordinates. They are 18 tuples with values (X, Y, C) for 2D and (X, Y, Z, C) for 3D; where C is the confidence score of the detected joint, X, Y and Z represent the coordinates on the X-axis, Y-axis and the Z-axis of the video frame, respectively.

1.2 Spatial Temporal Graph Neural Network

New techniques have been proposed recently to exploit the connections between the joints of a skeleton. Among these, Convolutional Neural Networks (CNNs) are used to address human action modelling tasks due to their ability to automatically capture the patterns contained in the spatial configuration of the joints and their temporal dynamics [10]. However, the skeletons are presented in graphs form-like, making it difficult to use conventional CNNs to model the dynamics of human actions. Thanks to the recent evolution of Graph Convolutional Neural Networks (GCNNs), it is possible to analyse the non-structured data in an end-to-end manner. These techniques generalize CNNs to the graph's structures [11]. It has been proven that GCNNs are highly capable of solving computer vision problems and have demonstrated superior performance as compared to CNNs approaches [12]. The remarkable success of GCNNs is based on the locally connected configurations and the collective aggregation upon graphical structures. Moreover, GCNNs operate on each node separately regardless of the input sequence. Meaning that, unlike CNNs, the outcome of GNNs is robust to changes in the input node information [13].

In order to achieve an accurate ADL recognition, the temporal dimension must be considered. An action can be considered as a time-dependent pattern of a set of joints in motion [8]. A graph offers a more intuitive representation of a skeleton by presenting the bones as edges and joints as vertices [14]. Given the advantages of GCNNs mentioned previously, numerous approaches for skeleton-based action recognition using this architecture have been proposed. The first GCNN-based solution for action recognition using skeleton data was presented by Yan et al. [6]. They considered both spatial and temporal dimensions of skeleton joints movements at the modelling stage. This approach is called the Spatiotemporal Graph Convolutional Network (ST-GCN) model. In the ST-GCN model, every joint has a set of edges for the spatial and temporal dimensions independently, as it is illustrated in Fig. 1. Suppose a given sequence of frames with skeleton joints coordinates; then the spatial edges connect each joint with its neighbourhood per frame. On the other hand, temporal boundaries connect each joint with another joint corresponding to the exact location from a consecutive frame. Meaning that, the temporal edge set represents the joint trajectory over time [6]. However, the topology of the graph is not implicitly structured like Euclidean-based data. For instance, most of the nodes have different numbers of neighbours. Therefore, multiple strategies for applying the convolution operation upon skeleton joints have been proposed.

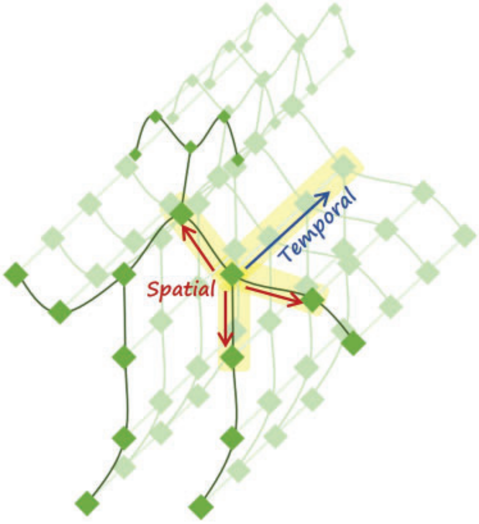

Figure 1: Spatiotemporal graph representation of a skeleton

In their work, Yan et al. [6] presented multiple solutions to perform the convolution operation over the skeleton graph. They first divided the skeleton graph into a fixed subset of nodes (the skeleton joints) they called *neighbour sets*. Every neighbour set has a central node (the *root node*) and its adjacent nodes. Subsequently, it is performed a partitioning of the neighbour set into a fixed number of K subsets, where a numeric label (which we call *priority*) is assigned to each of them. Formally, each adjacent node u_{ti} in a neighbour set $B(u_{tj})$ of a root node u_{tj} is mapped to a label l_{ti}. On the other hand, each filter of the CNN has a K number of subsets of values. Therefore, each subset of values of a filter performs the convolution operation process upon the feature vector of its corresponding node. Given that the skeleton data has been obtained using the Open Pose toolbox [9], each feature vector consists of the 2D coordinates of the joints, including a value of confidence C. These ideas are illustrated in Fig. 2.

In Fig. 2, two of the neighbour sets are shown with an orange background. Subsequently, the features of each of the nodes (x, y, c) are then concatenated into a feature matrix. However, the criteria to define the position of each of the feature vector in the final matrix is then defined by the utilized partition strategy. Amidst the skeleton partitioning strategies to perform the label mapping presented in [6], the *Spatial Configuration Strategy* served as a reference for the techniques proposed in the present study.

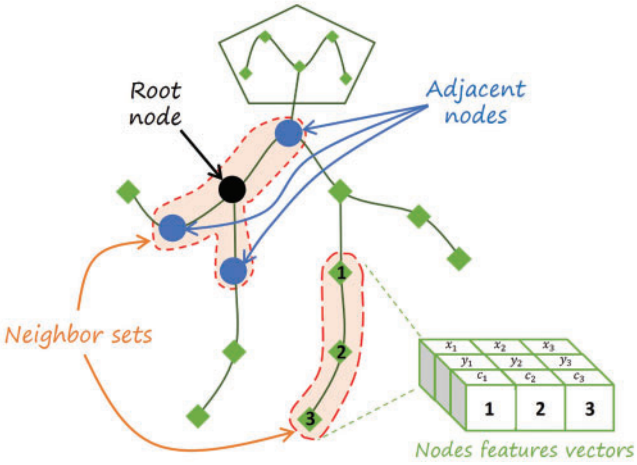

Figure 2: Skeleton components

1.2.1 Spatial Configuration Partitioning Strategy

In this strategy, the partitioning for the label mapping is performed according to the distance of each node in the neighbour set with respect to the centre of gravity *cg* of the skeleton graph. The *cg* is defined as the average of the coordinates of all the joints of the skeleton in a single videoframe [6]. According to [6], each neighbour set is divided into three (filter size $K = 3$). Therefore, each kernel has three subsets of values; one for the root node, one for the joints closer to *cg* and another one for the joints located farther with respect to *cg*. As it can be seen in Fig. 3, each filter with three subsets of values is applied to the node feature vectors in order to create the output feature map.

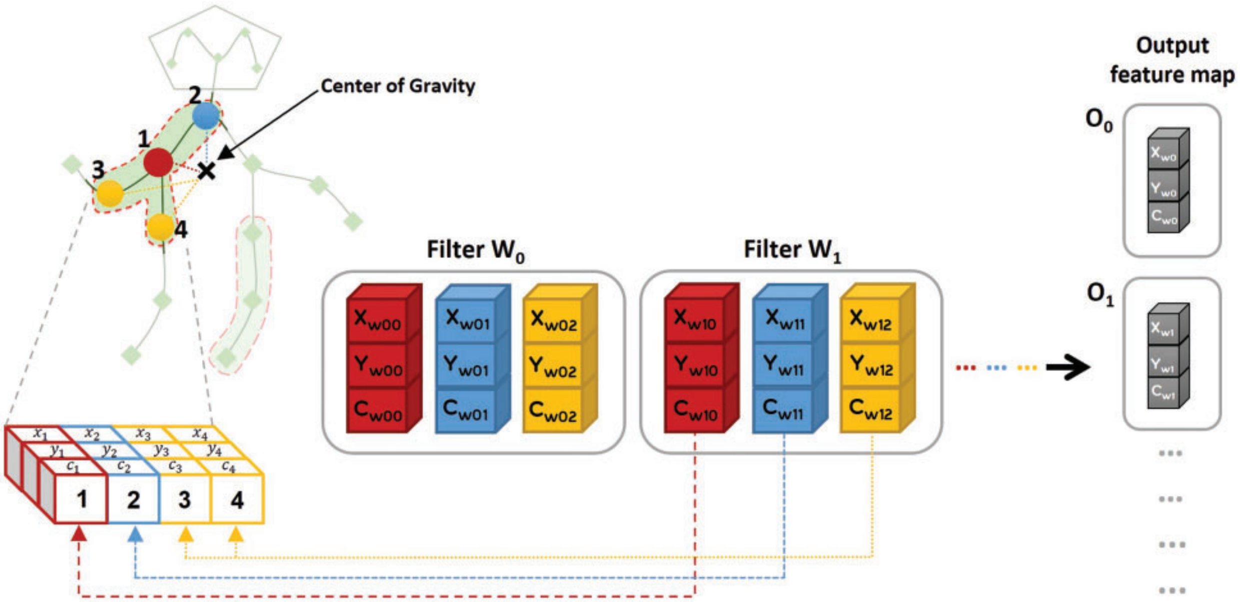
Center of Gravity
1
2
3
4
Filter W_0
X_{w00} Y_{w00} C_{w00}
X_{w01} Y_{w01} C_{w01}
X_{w02} Y_{w02} C_{w02}
Filter W_1
X_{w10} Y_{w10} C_{w10}
X_{w11} Y_{w11} C_{w11}
X_{w12} Y_{w12} C_{w12}
Output feature map
O_0
X_{w0} Y_{w0} C_{w0}
O_1
X_{w1} Y_{w1} C_{w1}
x_1 x_2 x_3 x_4
y_1 y_2 y_3 y_4
c_1 c_2 c_3 c_4
1 2 3 4

Figure 3: Spatial configuration partitioning

In this technique, the filter size $K = 3$, and the mapping are defined by the following [6]:

$$l_{ti}(u_{tj}) = \begin{cases} 0 & \textit{if } r_j = r_i \\ 1 & \textit{if } r_j < r_i \\ 2 & \textit{if } r_j > r_i \end{cases} \tag{1}$$

where l_{ti} presents the label map for each joint i in the neighbour set of the root node u_{tj}, r_i is the average distance from cg to the root node u_{tj} over each frame and r_i is the average distance from cg to the i_{th} joint over each frame across all the training set. Once the labelling of each node in the neighbour set has been set, the convolution operation is performed to produce the output feature maps, as shown in Fig. 3.

1.3 Learnable Edge Importance Weighting

It is important to note that complex movements can be inferred from a small set of representatives' *bright spots* on the joints of the human body [15]. However, not all the joints provide the same quality and quantity of information regarding the movement performed. Therefore, it is intuitive to assign a different level of importance to every joint in the skeleton.

In the ST-GCN framework proposed by Yan et al. [6], the authors added a mask M (or M-mask) to each layer of the GCNN to express the importance of each joint. The mask applied scales the contribution of each joint of the skeleton according to the learned weights of the spatial graph network. Accordingly, the proposed M-mask considerably improves architecture's performance. Therefore, the M-mask is applied to the ST-GCN network throughout their experiments.

1.4 Our Contribution

This work proposes an improved set of label mapping methods for the ST-GCN framework by introducing three split techniques (full distance split, connection split, and index split) as an alternative approach for the convolution operation. It is based upon the ST-GCN framework proposed by Yan et al. [6]. Our results indicate that all our proposed split strategies outperform the baseline model. Furthermore, the proposed frameworks are more stable during training. Finally, our proposals do not require additional training parameters of the edge importance weighting applied by the ST-GCN model. This proves that our proposal can provide a more suitable solution for real-time applications focused on daily living recognition systems activities for indoor environments.

The contributions are summarized below:

I: We present an improved set of label mapping methods for the ST-GCN framework by introducing three split techniques (full distance split, connection split, and index split) as an alternative approach for the convolution operation.

II: Instead of the traditional way of extracting information from the skeleton without considering the relations between the joints, we exploit the relationship between the joints during the action execution to provide valuable and accurate information about the action performed.

III: We find that an extensive analysis of the inner skeleton joint information by partitioning the skeleton graph in the most number of pieces possible results in more accurate data.

IV: We propose split strategies that focus on capturing the patterns in the relationship between the skeleton joints by carefully analysing the partition strategies utilized to perform the movement modelling using the ST-GCN framework.

The rest of the paper is structured as follows: Section 2 presents state-of-the-art review for previous skeleton graph-based action recognition approaches. The details of the proposed skeleton partition strategies are presented in Section 3. Section 4 discuss the experimental settings we use to obtain the results. The results and discussion are presented Section 5. Finally, Section 6 concludes the paper.

2 Related Literature

There has been previous work on AR upon skeleton data. Due to the emergence of low-cost depth cameras, access to skeleton data has become relatively easy [16]. Therefore, there has been an increasing interest in using skeleton representations to recognize human activity in general. For the sake of being conscience, few most recent but relevant works are mentioned. Zhang et al. [17] combined skeleton data with machine learning methods (such as logistic regression) upon dataset benchmarks. They demonstrated that skeleton representations provide better performance in terms of accuracy than other forms of motion representations. In order to model the dependencies between joints and bones, Shi et al. [14] presented a variety of graph networks denominated Directed Acyclic Graph (DAG). Later, Cheng et al. [18] presented a shift CNN inspired method called Shift-GCN. Their approach aims to reduce the computational complexity of previous ST-GCN-based methods. The results showed the achievement of 10× less computational complexity. However, to the best of our knowledge, there have not been unique partition strategies proposed to enhance the performance of an AR using the ST-GCN model presented in [6].

3 Proposed Split Strategies

In this section, we present a new set of techniques to create the label mapping for the nodes in the neighbour sets of the skeleton graph. The techniques are modifications of the previously proposed spatial configuration partitioning presented in [6].

As the baseline model, a maximum distance of one node with respect to the root node defines the neighbour sets in the skeleton graph. However, every node in the neighbour set is labelled *separately* in every strategy presented in this section. Therefore, in every proposed approach, the filter size $K = 4$. For instance, consider a neighbour set consisting only of the root node with a single adjacent node. For this case, the third and fourth subsets values of the kernel are set by zeros. Each of the split strategies proposed is computed in each frame of a training video sample individually.

Fig. 4 illustrates our proposed partitioning strategy. As it can be seen, a different label mapping is assigned to each node in the neighbour set. Therefore, a different subset of values of each filter is applied to each joint feature vector. However, the bottleneck is defining each node's order (split criterion) in the neighbour set. We propose three different approaches to address this issue: full distance split, connection split, and index split. These proposals are shown in Fig. 5 and will be explained in the following sections.

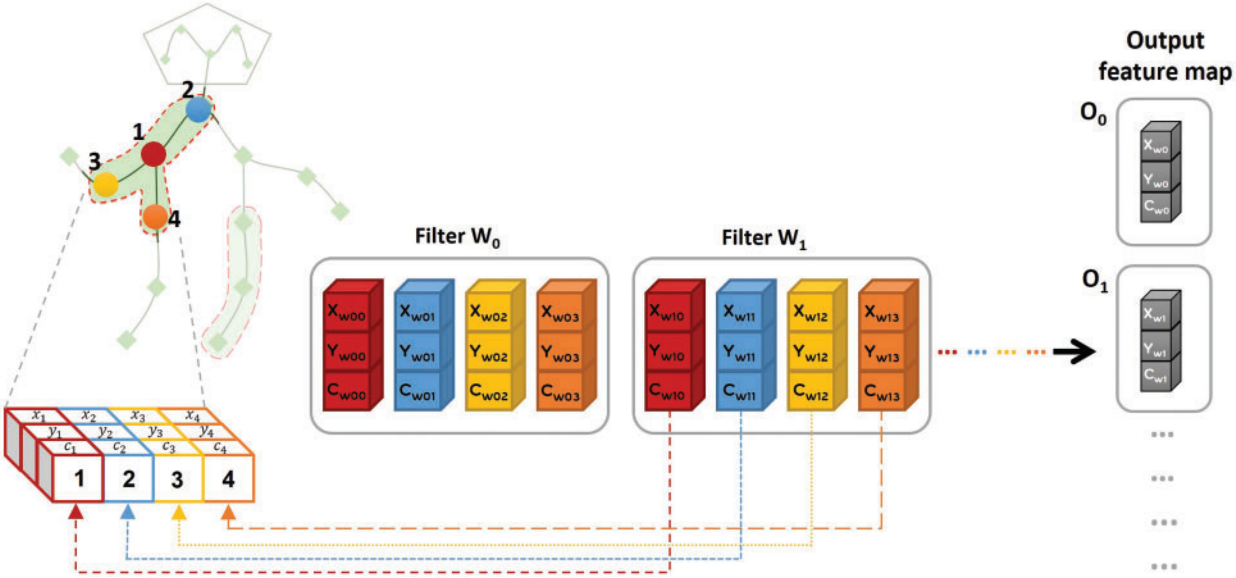

Figure 4: Proposed partition strategy

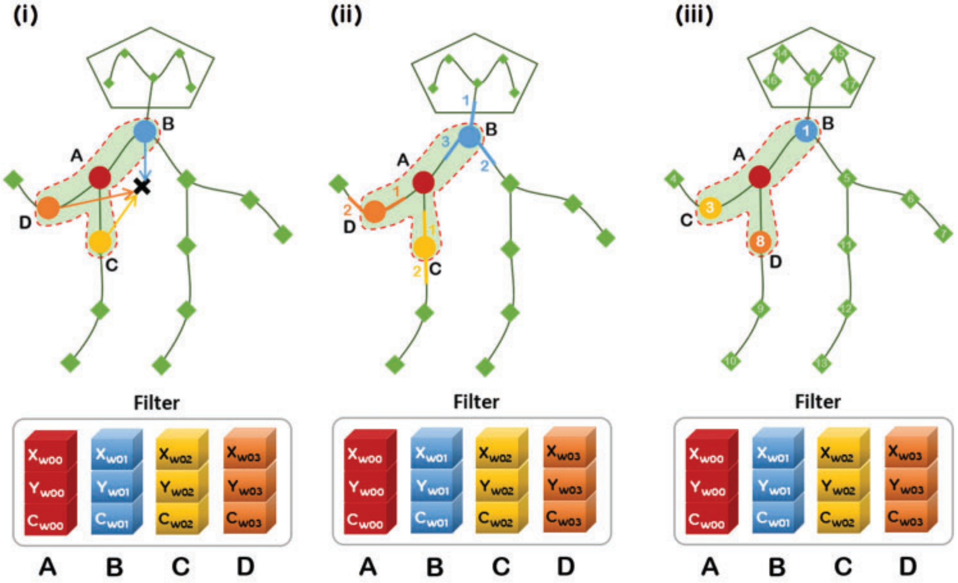

Figure 5: Proposed split techniques; (i) Full distance split. (ii) Connection split. (iii) Index split

3.1 Full Distance Split

In this method, the partitioning for the label mapping is performed according to the distance of each node in the neighbour set with respect to *cg*. As can be noticed, this solution is similar to the spatial configuration partitioning approach previously explained. However, here we consider the distance of every node in the neighbour set. Thus, this solution is named the *full distance* split technique. Therefore, depending on the neighbour set in the skeleton, each kernel can have up to four subsets of values. Fig. 5i shows that each filter with four subsets of values is applied to the node feature vectors. The order is defined by their relative distances with respect to *cg* to create the output feature map. To explain this strategy, we define the set *F* as the Euclidean distances of the i_{th} adjacent node u_{ti} (of the root node u_{tj}) with respect to *cg* sorted in ascending order as:

$$F = \{f_m | m = 1, 2, \ldots, \ N\} \tag{2}$$

where *N* is the number of adjacent nodes to the root node u_{tj}. For instance, f_1 and f_N have the minimum and maximum values in *F*, respectively. In this strategy, the label mapping is given by:

$$l_{ti}(u_{tj}) = \begin{cases} 0 & if\ u_{ti} - cg_2 = r_r \\ m & if\ u_{ti} - cg_2 = f_m \end{cases} \tag{3}$$

where l_{ti} represents the label map for each joint *i* in the neighbour set of the root node u_{tj}, x_r is the Euclidean distance from the root node u_{tj} to *cg*.

3.2 Connection Split

In this approach, the number of adjacent joints of each joint (i.e., the joint degree) represents the split criterion in the neighbor set. Thus, the more connections the joint has, the higher priority is assigned to it.

Fig. 5ii shows that the joint with label A represents the root node, and B is the joint with the highest priority since it has three adjacent joints connected. We observe that both C and D joints have two connections. Hence, the priority for these nodes is set randomly. Once the joint priorities have been set, the convolution operation is performed with a subset of values of each filter for every joint in the neighbor set independently.

To define the label mapping in this approach, we first define the neighbor set of a root node u_{tj} and *N* adjacent nodes as $B(u_{tj})$ [6], and we also define the degree matrix of $B(u_{tj})$ as *D*, where $D \in R^{NxN}$. Therefore, the values at the d_{ii} position of *D* contain the degree value $d(u_{ti})$ of the each of the adjacent nodes of the root node u_{tj}. Similarly, we define a set *C* as the degree values $d(u_{ti})$ of each of the *N* adjacent nodes of the root node sorted in descending order as follows:

$$C = \{c_m | m = 1, 2, \ldots, \ N\} \tag{4}$$

For instance, c_1 and c_N have the maximum and minimum values of *C*, respectively. Finally, the label mapping is thus defined as:

$$l_{ti}(u_{tj}) = \begin{cases} 0 & if\ d(u_{ti}) = d_r \\ m & if\ d(u_{ti}) = c_m \end{cases} \tag{5}$$

where l_{ti} represents the label map for each adjacent joint *i* to the root node u_{tj} in the neighbor set, and d_r is the degree corresponding the root node u_{tj}.

3.3 Index Split

The skeleton data utilized for our study is gathered using the Open-Pose [9] library. According to the library documentation, the output file with the skeleton information consists of critical/key points. The output skeleton provided by the Open Pose toolbox is shown in Fig. 6.

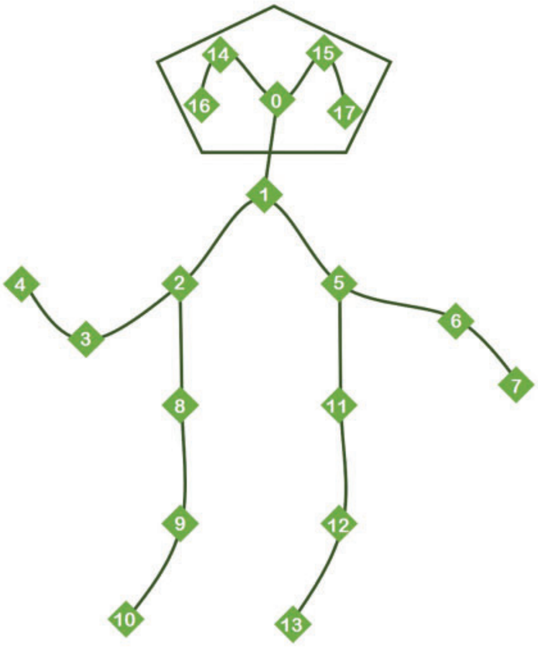

Figure 6: Open-Pose output key points

In this approach, the value of the index of each key point defines the priority criterion of the neighbour set. An illustrative example is shown in Fig. 5iii. For instance, joint B is assigned with the highest priority since it has a key point index value of 1, and C is the joint with the second priority since it has a key point index value of 3. Finally, D is the joint with the least priority since it has a key point index value of 8.

Therefore, we define the set P as the indexes of the key points $ind(u_{ti})$ of the i_{th} adjacent nodes u_{ti} (of the root node u_{tj}) sorted in ascending order as:

$$P = \{p_m | m = 1, 2, \ldots, N\} \tag{6}$$

where N is the number of adjacent nodes to the root node u_{tj}. For instance, p_1 and p_N have the minimum and maximum values of P, respectively. The label mapping is therefore defined as:

$$l_{ti}(u_{tj}) = \begin{cases} 0 & if\ ind(u_{ti}) = in_r \\ m & if\ ind(u_{ti}) = p_m \end{cases}, \tag{7}$$

where l_{ti} represents the label map for each joint i in the neighbour set of the root node u_{tj} and in_r is the index of the key point corresponding to the root node u_{tj}.

4 Experiments

4.1 Datasets

To evaluate the performance of our proposed partitioning techniques, we train our models on two benchmark datasets: the NTU RGB + D [19] and the Kinetics [20] dataset. These two datasets were considered in order to provide a valid comparison with the original ST-GCN framework.

4.1.1 NTU-RGB + D

Up to date, the NTU-RGB + D is known to be the most extensive dataset with 3D joints annotations for human AR tasks [6]. The samples have been recorded using the Microsoft Kinect V2 camera. In order to take the most advantage of the chosen camera device, each action sample consists of a depth map modality, 3D joint information, RGB frames, and infrared sequences. The information provided by this dataset consists of the tri-dimensional location of the 25 main joints of the human body.

In their study, Shahroudy et al. [19] proposed two evaluation criteria for the NTU-RGB + D dataset: the Cross-Subject (*X-sub*) and the Cross-View (*X-view*) evaluations. In the first approach, the train/test split for evaluation was based upon groups of subjects performing the action; the data corresponding to 20 participants is used for training and the remaining samples for testing. On the other hand, the X-view evaluation approach considers the camera view as criteria for the train/test split; the data collected by the camera 1 is used for testing and the data collected by the other two cameras is used for training.

The NTU-RGB + D dataset provides a total of 56,880 action clips performing 60 different actions classified into three major groups: daily actions, health-related actions, and mutual actions. Forty participants performed the test action samples. Each sample has been captured with 3 different cameras simultaneously located at the same height but different angles. Later, this dataset was extended twice its size by adding 60 more classes and another 57,600 video samples [19]. This extended version is called NTU RGB + D 120 (120-class NTU RGB + D dataset). By considering the 3D skeletons modality of the NTU-RGB + D dataset only, the storage was reduced from 136 GB to 5.8 GB. Therefore, the computational speed is reduced considerably.

4.1.2 Kinetics

While the NTU-RGB + D dataset is widely known to be the largest in-house captured AR dataset, the DeepMind Kinetics human action dataset is the largest set with unconstrained AR samples.

The 306,245 videos provided by the Kinetics dataset are obtained from YouTube. Each video sample is supplied with no previous editing to ensure good variable resolution and frame rate for action modelling and is classified into 400 different action classes.

Due to the vast quantity of classes, one video sample can be classified into more than one cluster. For instance, a video sample with a person texting while driving a car can be classified with the "texting" label or the "driving a car" label. Therefore, the authors in [20] suggest considering a top-5 performance evaluation rather than a top-1 approach. Meaning that, a labelled sample is considered a true positive if its ground truth label appears within the 5 classes with the highest scores predicted by the model (top-5); contrary to considering only the predicted class with the highest score (top-1).

The Kinetics dataset provides the raw RGB format videos. Therefore, it requires the skeleton information to be extracted from the sample videos. Accordingly, we use the dataset that contains the Kinetics-skeleton information provided by Yan et al. [6] for our experiments.

4.2 Model Implementation

The experiment process comprises of three stages: Data Splitting, ST-GCN model setup, and Model Training. These stages are explained as follows:

4.2.1 Data Splitting

The datasets are divided into two subsets: the training and the validation sets. In our experiments, we consider a 3:1 relation for training and validation split, respectively.

4.2.2 ST-GCN Model Setup

The ST-GCN model uses a baseline architecture. It consists of a stack of 9 layers that are divided into 3-layer blocks stacked together. Each layer block consists of 3 layers each. The layers of the first block have 64 output channels each. The second and third blocks have 128 and 256 output channels, respectively. Finally, the 256-feature vector output by the last layer is fed into a softmax classifier to predict the performed action [6].

4.2.3 Model Training

The ST-GCN model is implemented on the PyTorch framework for deep learning modelling [21]. The models are trained using stochastic gradient descent with learning rate decay as an optimization algorithm. The initial learning rate is 0.1. The number of epochs and decay schedule for training varies depending on the dataset used. For the NTU-RGB + D dataset, we train the models for 80 epochs, and the learning rate decays by a factor of 0.1 on the 10th and the 50th epochs. On the other hand, for the Kinetics dataset, we train the models for 50 epochs, and the learning rate decays by a factor of 0.1 every 10th epoch. Similarly, the batch size also varies according to the dataset utilized; for the NTU-RGB + D dataset, the batch sizes for training and testing used were 32 and 64, respectively; on the other hand, for the Kinetics dataset, the batch sizes for training and testing used were 128 and 256, respectively. To avoid overfitting, a weight decay value of 0.0001 has been considered. Additionally, a dropout value of 0.5 has been set for the NTU-RGB + D dataset experiments.

To provide a valid comparison with the baseline model, an M-mask implementation is considered in the experiments presented in this study.

5 Experimental Results and Discussion

This section discusses the performance of our proposals against the benchmark ST-GCN models based on [6] using the spatial configuration partition approach. This strategy provides the best performance in terms of accuracy in [6]. Therefore, it has been chosen as a baseline to prove the effectiveness of the partition strategies introduced in this study.

5.1 Results Evaluation on NTU-RGB + D

Note that we aim to recognize ADL in an indoor environment. Therefore, the NTU-RGB + D dataset serves as a more accurate reference than the Kinetics dataset since it was recorded using the same conditions. Hence, we focus on the results obtained with this dataset. We use the 3D joint information provided in [19] in our experiments. The Tab. 1 shows the performance comparisons of our proposals and the state-of-the-art ST-GCN framework. It can be observed that all our partition strategies outperform the spatial configuration strategy of the ST-GCN. For the X-sub benchmark, the connection split achieves the highest performance of 82.6% accuracy, more than 1% higher than the ST-GCN performance. On the other hand, the index split outperforms the rest of the strategies with 90.5% accuracy on the X-view benchmark, more than 2% higher than the ST-GCN performance.

Table 1: NTU-RGB + D performance

	Method	X-sub	X-view
ST-GCN	Spatial configuration partitioning	81.5%	88.3%
Ours	Full distance split	81.6%	89.3%
Ours	Connection split	**82.6%**	89.6%
Ours	Index split	81.7%	**90.5%**

Figs. 7–10 show the training behaviour of the models using the spatial configuration partitioning of the ST-GCN framework and the proposed connection split on both X-sub and X-view benchmarks without the M-mask implementation. The blue and orange plots show the performance of the models using the training and the validation sets, respectively. The training score plots show that the learning performance of the proposed connection split stabilizes while increasing over time compared with the ST-GCN outcome. Our proposals provide a considerable advantage over the benchmark framework because it demonstrates that the M-mask is not required to yield satisfactory performance. The omission of the M-mask results in a reduction of computational complexity. Hence, our proposal can provide a more suitable solution for real-time applications. Moreover, given the performance superiority on accuracy and time consumption, our proposed method offers a practical solution an ADL recognition system.

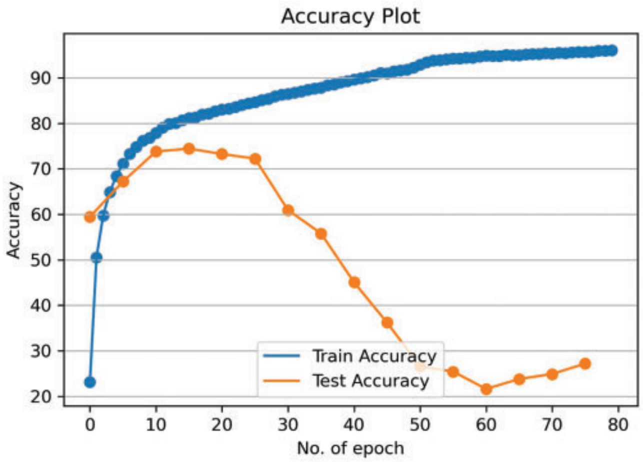

Figure 7: Spatial C.P X-sub training scores

5.2 *Performance on the Kinetics Dataset*

The recognition performance has been evaluated using the top-1 and top-5 criterion using the Kinetics dataset. We validate the performance of our proposed techniques with the ST-GCN framework, as shown in Tab. 2.

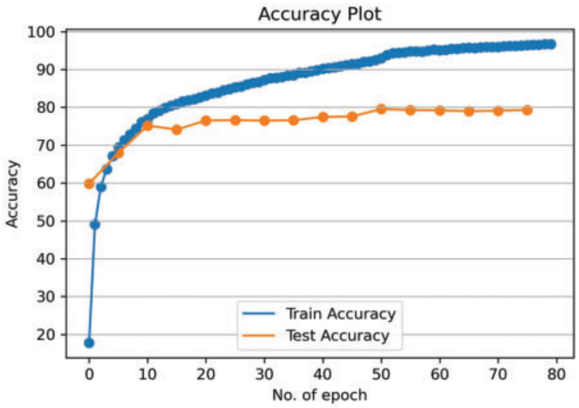

Figure 8: Connection split X-sub training scores

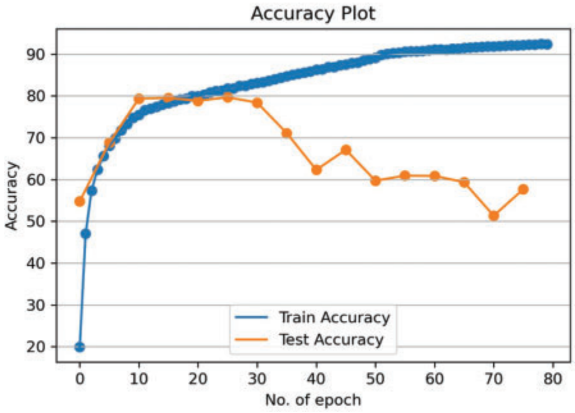

Figure 9: Spatial C.P X-view training scores

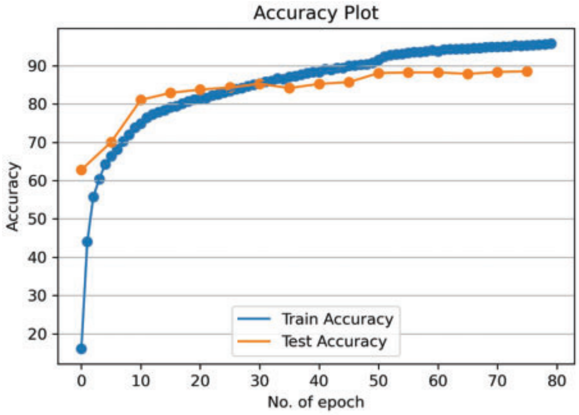

Figure 10: Connection split X-view training scores

Table 2: Performance on kinetics dataset

	Method	Top-1	Top-5
ST-GCN	Spatial configuration partitioning	30.7%	52.8%
Ours	Full distance split	**31.7%**	**54.5%**
Ours	Connection split	30.7%	53.3%
Ours	Index split	31.5%	54.1%

As the results indicate, all our partition strategies outperform the spatial configuration strategy of the ST-GCN using the top-5 criteria. We observe that 54.5% accuracy is achieved using the full distance split approach, which is 2% higher than the performance obtained with the baseline model. On the other hand, by using the top-1 evaluation criteria, our proposal achieves the same performance as the ST-GCN model. Similarly, using this evaluation basis, the highest performance achieved is a 31.7% accuracy using the full distance split approach resulting in a 1% margin higher than the result obtained with the ST-GCN model.

Therefore, we can conclude that the performance metrics presented in Tab. 2 validates the superiority of the full distance split method proposed on the Kinetics dataset.

6 Conclusion

In this work, we propose an improved set of label mapping methods for the ST-GCN framework (full distance split, connection split, and index split) as an alternative approach for the convolution operation. Our results indicate that all our split techniques outperform the previous partitioning strategies for the ST-GCN framework. Moreover, they demonstrate to be more stable during training without using the additional training parameter of the edge importance weighting applied by the baseline model. Therefore, the results obtained with our current split proposals can provide a more suitable solution for real-time applications focused on ADL recognition systems for indoor environments than the baseline strategies for the ST-GCN framework.

A significant computational effort is involved in using heterogeneous methods to calculate the distances between the joints and the *cg* for each frame in the video sample for full distance split and spatial configuration partitioning. It will be computationally less demanding to use a homogeneous technique to calculate the distance between the joints and the *cg* for both splitting strategies. Furthermore, while our current methodology considers greater distances from the root node to perform the skeleton partitioning, additional flexibility can be made by increasing the amount joints per neighbour set. This may give room to cover larger body sections (such as limbs), making it possible to find more complex relationships between the joints during the execution of the actions.

Acknowledgement: The authors acknowledge the support of King Abdulaziz City of Science and Technology.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Prati, C. Shan and K. Wang, "Sensors, vision and networks: From video surveillance to activity recognition and health monitoring," in *Journal of Ambient Intelligence and Smart Environments*, vol. 11, no. 1, pp. 5–22, 2019.

[2] E. Casilari, M. Álvarez-Marco and F. García-Lagos, "A study of the use of gyroscope measurements in wearable fall detection systems," in *Symmetry*, vol. 12, no. 4, pp. 649, 2020.

[3] K. de Miguel, A. Brunete, M. Hernando and E. Gambao, "Home camera-based fall detection system for the elderly," in *Sensors*, vol. 17, no. 12, pp. 2864, 2017.

[4] K. Kinoshita, M. Enokidani, M. Izumida and K. Murakami, "Tracking of a moving object using one-dimensional optical flow with a rotating observer," in *9th Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, pp. 1–6, 2006.

[5] A. F. Bobick, "Movement, activity and action: The role of knowledge in the perception of motion," in *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 352, no. 1358, pp. 1257–1265, 1997.

[6] S. Yan, Y. Xiong and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *the Thirty-Second AAAI Conf. on Artificial Intelligence*, New Orleans, Louisiana, USA, pp. 7444–7452, 2018.

[7] H. Foroughi, B. S. Aski and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," in *11th Int. Conf. on Computer and Information Technology*, Khulna, Bangladesh, pp. 219–224, 2008.

[8] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang *et al.,* "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, California, USA, pp. 3595–3603, 2019.

[9] Z. Cao, T. Simon, S. E. Wei and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA, pp. 1302–1310, 2017.

[10] J. Tu, M. Liu and H. Liu, "Skeleton-based human action recognition using spatial temporal 3d convolutional neural networks," in *IEEE Int. Conf. on Multimedia and Expo*, San Diego, CA, USA, pp. 1–6, 2018.

[11] C. Si, W. Chen, W. Wang, L. Wang and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, California, USA, pp. 1227–1236, 2019.

[12] S. Zhang, H. Tong, J. Xu and R. Maciejewski, "Graph convolutional networks: A comprehensive review," in *Computational Social Networks*, vol. 6, no. 1, pp. 11, 2019.

[13] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu *et al.,* "Graph neural networks: A review of methods and applications," in arXiv:1812.08434. [Online]. Available: https://arxiv.org/abs/1812.08434, 2019.

[14] L. Shi, Y. Zhang, J. Cheng and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, California, USA, pp. 7904–7913, 2019.

[15] G. Johansson, "Visual perception of biological motion and a model for its analysis," in *Perception & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[16] R. Vemulapalli, F. Arrate and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 588–595, 2014.

[17] Z. Zhang, J. Sun, J. Zhang, Y. Qin and G. Sun, "Constructing skeleton for parallel applications with machine learning methods," in *Proc. of the 48th Int. Conf. on Parallel Processing: Workshops*, Kyoto, Japan, pp. 1–8, 2019.

[18] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng *et al.,* "Skeleton-based action recognition with shift graph convolutional network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 180–189, 2020.

[19] A. Shahroudy, J. Liu, T. T. Ng and G. Wang, "NTU RGB + D: A large scale dataset for 3d human activity analysis," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, pp. 1010–1019, 2016.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier *et al.,* "The kinetics human action video dataset," in arXiv:1705.06950. [Online]. Available: https://arxiv.org/abs/1705.06950, 2017.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury *et al.,* "Pytorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, Vancouver, Canada: Curran Associates Inc., pp. 8024–8035, 2019.