DeGCN: Deformable Graph Convolutional Networks for Skeleton-Based Action Recognition

Woomin Myung*, Nan Su*, Jing-Hao Xue, *Senior Member, IEEE,* and Guijin Wang, *Senior Member, IEEE*

***Abstract*—Graph convolutional networks (GCN) have recently been studied to exploit the graph topology of the human body for skeleton-based action recognition. However, most of these methods unfortunately aggregate messages via an inflexible pattern for various action samples, lacking the awareness of intra-class variety and the suitableness for skeleton sequences, which often contain redundant or even detrimental connections. In this paper, we propose a novel Deformable Graph Convolutional Network (DeGCN) to adaptively capture the most informative joints. The proposed DeGCN learns the deformable sampling locations on both spatial and temporal graphs, enabling the model to perceive discriminative receptive fields. Notably, considering human action is inherently continuous, the corresponding temporal features are defined in a continuous latent space. Furthermore, we design an innovative multi-branch framework, which not only strikes a better trade-off between accuracy and model size, but also elevates the effect of ensemble between the joint and bone modalities remarkably. Extensive experiments show that our proposed method achieves state-of-the-art performances on three widely used datasets, NTU RGB+D, NTU RGB+D 120, and NW-UCLA.**

***Index Terms*—Skeleton-based action recognition, graph convolutional network, deformable convoltuion.**

I. Introduction

HUMAN action recognition plays a significant role in many applications, including human-computer interaction and video surveillance [1]–[3]. In recent years, different from the conventional approaches that use RGB video for input, skeleton-based human action recognition has received much attention due to its compact representations and robustness to background changes.

Early-stage approaches to skeleton-based action recognition mainly focus on designing hand-crafted features [4], [5], which cannot consider various characteristics of human action simultaneously. With the development of deep learning, many data-driven methods based on recurrent neural networks (RNN) [6]–[8] or convolutional neural networks (CNN) [9], [10] have been proposed, but they structure the skeleton data as a vector sequence or a pseudo-image. These approaches are challenging to model the graph topology of the human body, which is essential for accurately understanding human actions, resulting in unsatisfactory recognition performance.

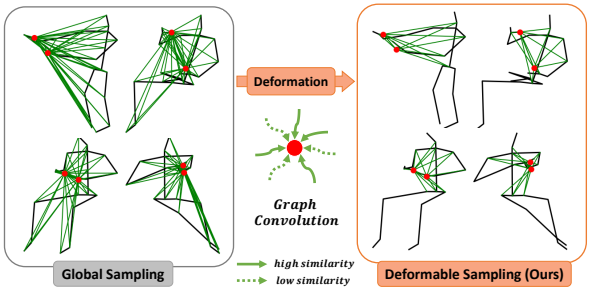


Fig. 1: Comparison between popular global graph convolution ($left$) and our **deformable graph convolution** ($right$) methods on spatial graphs with four different samples of "$reading$". The colored lines denote the sampling locations of the graph convolution. Our method adaptively samples the most relevant joints based on their similarity.

Recently, graph convolutional network (GCN) [11] has become one of the most popular approaches for skeleton-based action recognition due to its superiority in leveraging the structural information of graphs. Fundamentally, there are two kinds of GCNs to process graph data: spectral and spatial. Spectral methods [5]–[9] achieve convolution in the Fourier domain by mapping convolutional filters and graphs using the eigen-decomposition of their Laplacians. For instance, MLGCN [12] learns convex combinations of Laplacians, each one dedicated to a particular setting of the manifold enclosing the graph data. Additionally, learning convolutional filters and the Laplacian operators [13], embedded in a Chebyshev basis, was validated to help increase the discrimination power of graph representation. While these spectral methods make convolutions well defined, the learned filters rely on the Fourier basis ($i.e.$, Laplacian eigenbasis), making the model graph-dependent and less adaptable to highly varied topologies. Moreover, they require solving an eigen-decomposition of the Laplacian, which is both computationally expensive and sensitive to intra-class variability [14], [15]. In contrast, spatial methods [16]–[22] achieve convolution in the input domain without any preliminary step of spectral decomposition. They mainly rely on message passing, via attention matrices that

Manuscript created March 20.

Woomin Myung, Nan Su are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: auddnals@gmail.com; sunan@tsinghua.edu.cn).

Jing-Hao Xue is with the Department of Statistical Science, University College London, London WC1E 6BT, U.K (e-mail: jinghao.xue@ucl.ac.uk).

Guijin Wang is with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China, and also with the Shanghai Artificial Intelligence Laboratory, 200232, Shanghai, China (*Corresponding author*, e-mail: wangguijin@tsinghua.edu.cn).

* Authors contributed equally to this work.

capture context and connectivity between nodes, deeming these methods more effective compared to spectral one. ST-GCN [16] is the first work to apply spatial GCN to a skeleton-based action recognition task, which aggregates messages from its neighbor joints along the natural connectivity of the human body. However, it is difficult to capture correlations between non-directly connected joints ($e.g.$, two hands) in this fixed hand-crafted topology. For this reason, the key of many recent works [17]–[20], [22]–[25] lies in how to adaptively capture the potential correlations between joints.

However, most of these graph convolution algorithms still learn the representations of each joint via an inflexible message-passing scheme for various human actions. For example, Li *et al.* [17] exploit the higher-order polynomial of predefined adjacency matrices to expand the receptive field of graph convolutions. In this case, since the messages are aggregated by the joints from local body parts, it is ineffective to obtain information from distant joints. On the other hand, Shi *et al.* [18] construct a complete skeleton graph to capture long-range dependencies by setting the adjacency matrix as learnable parameters. Chen *et al.* [22] learn pairwise correlations to model channel-wise topologies. Sahbi [15] explores various constraints, such as orthogonality and stochasticity, acting as regularizers on learned matrix operators to more effectively learn topological properties. However, their representation capability is still limited due to high intra-class variances of human actions. For example, a person can perform the same action of "$reading$" in quite different ways, including standing, sitting, or lying down, as shown in Fig. 1($left$). In this case, the negatively correlated joints with high variances can lead to different representations even within the same action category, especially for the action samples where most of the joints are non-informative ($e.g.$, distinguishing "$writing$" and "$typing$"). This performance bottleneck raises an important question: why not adaptively select the most informative joints for message passing in graph convolution? The simple answer is: the skeleton graph is inherently discrete, and the hard selection process is non-differentiable, causing difficulties in end-to-end learning.

In this paper, we propose a novel deformable graph convolutional network (DeGCN) for skeleton-based action recognition. Our contributions are three-fold: (i) We propose a deformable spatial graph convolution (DeSGC) module to adaptively capture the most relevant joints. We sample only k joints with the highest similarity to each joint as neighbors when performing the graph convolutions (Fig. 1($right$)). In other words, the non-informative joint nodes intervening in message passing are adaptively eliminated. Notably, the similarity is used only for sampling operations, and the weight for message passing of sampled joints is designed with a separate pathway, allowing more focus on each task. (ii) We design a deformable temporal graph convolution (DeTGC) module to obtain dynamic and continuous receptive fields on temporal graphs. The key insight is to set the sampling locations as data-driven learnable parameters. Since the locations are real numbers rather than integers, we extract corresponding frame features via interpolation. (iii) We present a novel multi-branch framework that includes temporal scale-wise modeling (TSM) and a joint-bone fusion (JBF) stream to achieve a better trade-off between accuracy and model size. Our experiments demonstrate that our framework brings significant performance boosts with a comparable number of parameters.

Learning the sampling process for both spatial and temporal skeleton graphs in a differentiable way, without rigidly modeling topology ($i.e.$, receptive field of the kernel) that may include non-informative joints, distinguishes this model from all the aforementioned related work. This allows the model to accommodate intra-class variations and better understand fine-grained representations through end-to-end learning. To quantitatively verify the effectiveness of our DeGCN, we conduct extensive experiments and benchmark our results against competitive baselines on three widely used skeleton-based action recognition datasets, NTU RGB+D [26], NTU RGB+D 120 [27] and NW-UCLA [28]. Extensive experimental results show that the proposed DeGCN achieves state-of-the-art performances on these three datasets.

II. Related Work

A. GCN-based Action Recognition

In recent years, GCN-based methods [16], [18]–[20], [22], [23], [25], [29]–[31] for skeleton-based action recognition have shown significant performance boosts compared with other methods, by capturing more semantic relationships between joints. Specifically, numerous graph convolution algorithms are developed based on two approaches: $local\ message\text{-}passing$ [16], [17], [19], [20] and $global\ message\text{-}passing$ [18], [22], [23], [29], [31]. The $local\ message\text{-}passing$ approach mainly focuses on designing predefined topologies that aggregate messages from local neighborhoods. On the other hand, the $global\ message\text{-}passing$ approach constructs a fully-connected skeleton graph by correlation modeling, which typically has more robust recognition performance than the former due to dynamic topologies. However, considering that different human action samples often have different informative joints, it is not necessary to rigidly aggregate all joints, especially non-informative joints. In contrast, our approach can adaptively samples and aggregates the most informative joints for each action sample (See Fig. 1).

The previous work [32], named AdaSGN, closest to our work addresses the problem of eliminating non-informative joints. AdaSGN first pretrains several data-driven transform matrices with different numbers of joints for downsampling on the spatial dimension. The corresponding transformed features are then passed through a spatial modeling module and fed into a policy network, which selects the optimal joint number. By contrast, our DeGCN adaptively explores the most informative joints instead of specifying the number of joints, by taking into account the sample-wise correlations between joints. This increases the flexibility of the model for various action samples. Moreover, our deformable graph convolution algorithm is devised not only for spatial but also for temporal graphs, making the action sequences continuous in a latent space.

B. Deformation Modeling

In the early stage of deep learning, convolutional neural networks (CNNs) [33] have achieved significant success in various computer vision tasks, such as image classification [34], and object detection [35]. However, it is difficult to accommodate geometric variations into their fixed hand-crafted structures. To address this issue, a number of works for deformation modeling have been proposed, and they generally focus on how to adaptively direct important locations on images. Dai *et al.* [36] learn the 2D offsets from preceding feature maps, and add them to the regular grid sampling locations. Zhu *et al.* [37] propose Deformable-DETR that combines the advantages of deformable convolution [36] and DETR [38]. DPT [39] designs a plug-and-play module that learns the offsets and scales of each patch, and integrates it into Pyramid Vision Transformer [40]. However, these continuous offsets cannot be applied directly to inherently discrete data structures, such as skeleton graphs. In contrast, our approach can adaptively explore the most informative joint nodes for both spatial and temporal graphs of skeleton action sequences.

Recently, Park *et al.* [41] apply deformation modeling to the graph data structure to perform deformable convolution in multiple latent spaces. They first select neighbor nodes by generating several kNN graphs corresponding to different numbers of hops. The deformation is then performed by adding the offset vectors using an MLP network in a latent space. Obviously, this approach is not suitable for applying to our task, since it is difficult to capture the correlations of structurally distant but semantically important joints ($e.g.$, two hands). In contrast, our approach directly specifies the most informative neighbor joints with semantic information of the human body in the global skeleton graph, by a differentiable sampling process. In other words, we can simultaneously perform selection of neighbor joints and deformation of receptive fields. It is more flexible and intuitive. Furthermore, our deformable graph convolution algorithm is devised not only for spatial but also for temporal graphs.

In addition, the kernel-based approach [21] maps graph signals from an input space into a high-dimensional Hilbert space using kernel function. This implicit mapping can enhance the discrimination power of the graph representations and also adapt the receptive field without explicitly realigning nodes, thereby making it permutation-agnostic. While the experimental results have demonstrated its outstanding performance, modeling the kernel-based graph convolution requires a careful design. By contrast, our approach is not restricted by the constraints associated with a kernel function, including considerations such as neural consistency, yet it maintains the capability to dynamically adjust the receptive field.

III. Design of DeGCN

In this section, we elaborately introduce a novel framework, Deformable Graph Convolutional Networks, named DeGCN. In particular, four main improvements are made in our approach, $i.e.$, Deformable Spatial Graph Convolution (DeSGC) module, Deformable Temporal Graph Convolution (DeTGC) module, Temporal Scale-wise Modeling (TSM), and Joint-Bone Fusion (JBF) stream. We give a coarse-to-fine (from network to module) overview below.

A. Preliminaries

1) Notations: A human skeleton graph is represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, ..., v_N\}$ is the set of N joints, and $\mathcal{E}$ is the set of intra-skeleton and inter-frame edges. Suppose that the input skeleton data is $\mathbf{X} \in R^{C\times T\times N}$, where C and T denote the 3D Cartesian coordinates and the number of frames of the skeleton sequence, respectively.

2) Graph Convolutional Networks: Given the input data $\mathbf{X}$ and the output data $\mathbf{Y} \in R^{C'\times T\times N}$, where each joint v_i's features $\mathbf{x}_i \in R^{C\times T}$ and $\mathbf{y}_i \in R^{C'\times T}$, the vanilla spatial graph convolution operation widely used in skeleton-based action recognition [16] is formulated as

$$\mathbf{y}_i = \sigma\Big(\sum_{v_j\in\mathcal{N}(v_i)} \mathbf{W}_X \mathbf{x}_j a_{ij}\Big), \tag{1}$$

where $\sigma(\cdot)$ is an activation function and $\mathcal{N}(\cdot)$ is the set of neighbor joints, $\mathbf{W}_X \in R^{C'\times C}$ denotes a weight matrix for feature transformation, and a_{ij} is the correlation strength between v_i and v_j, an element of the normalized adjacency matrix $\mathbf{A} \in R^{N\times N}$.

B. Overview

The overall architecture of our method is presented in Fig. 2. On a high level, we adopt joint, bone, and velocity modalities as inputs and treat them separately via corresponding feature extraction streams, following recent studies [22], [29], [42]. Additionally, in our work, considering that there is an inherently natural connectivity between the joint and bone modalities, we capture their correlation through the proposed JBF stream. Consequently, our model has a three-input four-stream structure, and each prediction result is ensembled at the inference stage, as shown in Fig. 2(a).

In the following sub-sections, we describe each stream and its components in detail, including the feature extraction stream (Section III-C), the JBF stream (Section III-D) and the temporal scale-wise modeling (Section III-E).

C. Feature Extraction Stream

In each of our three feature extraction streams, which takes a single modality as input, a simple multi-branch form is conducted to learn diverse representations, as shown in Fig. 2(b). Each branch comprises an initial block for data-to-feature transformation and nine basic blocks ($i.e.$, $L = 9$) for extracting rich spatial-temporal representations, followed by a global average pooling layer and a fully connected layer. The initial block consists of a conventional graph convolution (GC) module implemented by Equation (1) and our temporal modeling (TM) module. The basic block replaces the GC module of the initial block with our spatial modeling (SM) module. Details of each modeling module are described in Section III-E and Section IV.

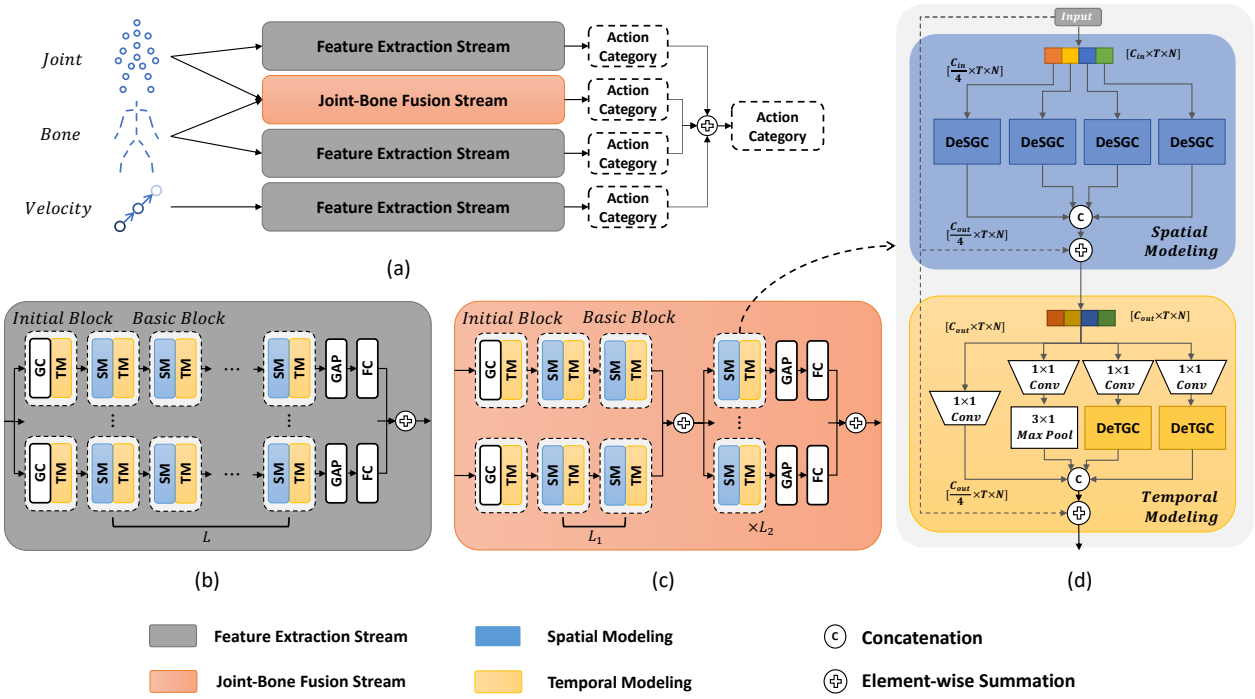


Fig. 2: Architecture overview of our proposed DeGCN. (a) is a multi-modal ensemble performed in our approach. (b) is the feature extraction stream, with single-modality as input. (c) is the proposed joint-bone fusion (JBF) stream, with *joint* and *bone* modalities as input. Note that the multi-branch form is conducted in both (b) and (c). (d) is the proposed temporal scale-wise modeling (TSM) consisting of spatial modeling (SM) and temporal modeling (TM) modules, in which different input colors represent different temporal scales. The dotted lines denote residual connection.

Notably, the learnable parameters of each branch are randomly initialized and are not shared between streams. Then, each stream is trained in an end-to-end manner by summing the cross-entropy losses of its multiple branches. We can obtain the prediction result via element-wise summation at the inference stage.

D. Joint-Bone Fusion Stream

The bone modality, which is first introduced in 2s-AGCN [18], is represented through the semantic connection of the human body. Along with the joints, it is an indispensable clue to constitute the skeleton graph. However, most existing methods ignore the correlation between these two modalities and treat them separately.

To address this issue, we further design a joint-bone fusion (JBF) stream in a mid-fusion manner, as shown in Fig. 2(c). In detail, we first extract discriminative representations through the stream consisting of an initial block and two basic blocks ($i.e.$, $L_1 = 2$) for each modality. We then fuse them via element-wise summation with a batch normalization layer. Subsequently, the fused output is passed through seven basic blocks ($i.e.$, $L_2 = 7$) to learn the correlated representations in a multi-branch form, similar to the feature extraction stream. Experimental results show that the proposed JBF stream can significantly boost the effect of ensemble between the joint and bone modalities (Section V-D).

E. Temporal Scale-wise Modeling

The multi-branch structure, which is conducted in both of the aforementioned two types of streams, is effective in improving recognition performance by learning various representations. However, it has a critical drawback that the model size increases linearly with the number of branches. Thus, we design a temporal scale-wise modeling (TSM) and apply it to basic blocks to reduce the redundant parameters and computations of each branch.

1) Spatial Modeling: A human action consists of several partial actions, which means that key joints in short-term and long-term actions can often be different. Inspired by this, our main idea is to learn the representations for each temporal scale individually. Specifically, we first split the channel of $\mathbf{X} \in R^{C_{in} \times T \times N}$ obtained from the previous temporal modeling module evenly to obtain the features $\{\mathbf{X}^{(s)}\}_{s=1}^{S}$ that corresponds to S temporal scales. We then perform our proposed deformable spatial graph convolution (DeSGC) for each $\mathbf{X}^{(s)}$ to learn temporal scale-specific spatial representations, and the extracted representations are concatenated, as shown in the spatial modeling module of Fig. 2(d). Note that we set the output channel sizes of each DeSGC module to $1/S$, to maintain the original size, and we also use temporal scale-wise residual connections to facilitate training.

2) Temporal Modeling: To extract action representations with different durations, we adopt the multi-scale temporal modeling module following [22]. Considering that our proposed deformable temporal graph convolution (DeTGC) module can adaptively learn the discriminative receptive field, as we explain in Section IV-B, we replace each vanilla temporal convolution with the DeTGC modules, as shown in the temporal modeling module of Fig. 2(d). Each branch contains a 1×1 convolution for channel reduction.

3) Discussion: Different from the most existing methods [16], [18], [20], [22], [23], our proposed TSM independently learns the spatial representations for each temporal scale. It can be seen that the extracted multi-scale representations are fused by 1×1 convolution layers of the following temporal modeling module without additional parameters. Hence, considering most existing spatial graph convolution approaches, including our DeSGC, consist of several point-wise convolutions, applying our TSM can reduce the parameters for modeling spatial relations by almost S times. Furthermore, this application can be extended to any method of adopting multi-scale temporal modeling modules (See Section V-E). In the next section, we describe how to extract spatial-temporal representations with the DeSGC and DeTGC modules.

IV. Deformable Graph Convolution

Human actions usually involve high intra-class variances (*e.g.*, Fig. 1), which actually motivated us to focus more on the joints with important meanings for action recognition to accommodate these variations and understand the fine-grained representations better. Moreover, this becomes more prominent as the graph grows larger along the temporal dimension. Therefore, one of the desirable characteristics of a robust GCNs-based algorithm for skeleton-based action recognition is the ability to aggregate messages by capturing the most relevant joints. In this section, we present the DeSGC and DeTGC modules to perform deformable graph convolutions on spatial and temporal graphs, respectively, and discuss their advantages over previous works.

A. Deformable Spatial GC (DeSGC)

Different from the grid data such as images, the skeleton graph is not globally continuous in the spatial dimension, which means that the indices of adjacent joints may not be adjacent. In this case, the negatively correlated joints with high variances can lead to different representations even within the same action category, especially for the action samples where most of the joints are non-informative (*e.g.*, distinguishing "*writing*" and "*typing*"). In this sub-section, we first describe how to perform the key joint selection on spatial graphs. Here, a calibration offset is introduced to make the sampling operation differentiable. We then describe the corresponding aggregation pathway, which is designed separately from the selection pathway. The architecture of the proposed DeSGC module is shown in Fig. 3(a).

1) Key Joint Selection Pathway: In essentially discrete data such as spatial skeleton graph, the first problem to consider is which joint should be sampled as neighbors given the center joint $\mathbf{x}_c \in R^{C_{in} \times T}$ of the kernel. To this end, one intuitive approach is to sample the neighboring joints based on the strength of their semantic connections for each action sample. Indeed, several recent works [18], [24], [43] show that calculating the similarity between two joints is one of the effective methods for determining whether there is a semantically connection between two joints and representing how strong the connection is. Thus, we also measure the similarity between the two joint features $\mathbf{x}_i$ and $\mathbf{x}_j$ as

$$\pi_{ij} = a_{ij} + \alpha \cdot \sigma\left(\frac{1}{T}\sum_{t}^{T} \langle \phi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle\right), \tag{2}$$

where a_{ij} denotes the adjacency matrix component, α is a learnable scalar for balancing the intensity of sample-specific similarity, $\sigma(\cdot)$ denotes activation function, $\langle \cdot, \cdot \rangle$ is normalized inner-product operation, $\phi(\cdot)$ and $\psi(\cdot)$ denote linear projection functions. Instead of performing the inner-product with C_{in} channels, $\mathbf{x}_i$ and $\mathbf{x}_j$ are linearly projected multiple times using different projection functions (*i.e.*, multi-head mechanisms) to capture various correlations, and both of these functions are chosen as the 1×1 convolutional layers, following [44]. The first term in Equation (2) is the data-driven similarity, and the second term is the temporal average of the sample-specific similarities obtained by applying self-attention [44]. Given the similarities, our goal is to sample only k joints with the highest similarity and define them as neighbors of $\mathbf{x}_c$. However, if the sampling operation is implemented in a conventional non-differentiable algorithmic manner, the similarity π_{ij} cannot be trained using prevalent gradient descent algorithms. Therefore, we approximate the one-hot vectors corresponding to the *top-k* indices to design a differentiable sampling strategy.

2) Calibration Offset: In general, a joint is correlated simultaneously with several other joints, which means that the similarities between joints follow a multimodal distribution, as shown in Fig. 3(b). In this case, to obtain differentiable one-hot vectors of *top-k* indices, we construct k well-calibrated probability distributions by adding a weighted one-hot offset to each of the indices and taking a softmax function. We define the m-th *top-k* well-calibrated probability distribution of a joint v_i as $\{\widehat{\pi}_{ij}^{(m)}\}_{j=1}^{N}$, where the probability $\widehat{\pi}_{ij}^{(m)}$ is formulated as

$$\widehat{\pi}_{ij}^{(m)} = \frac{\exp(\pi_{ij} + \Delta\widehat{\pi}_{ij}^{(m)})}{\sum_{j=1}^{N} \exp(\pi_{ij} + \Delta\widehat{\pi}_{ij}^{(m)})}. \tag{3}$$

Note that the calibration offset $\Delta\widehat{\pi}_{ij}^{(m)} = \delta$ if $j \in \{p_{i1}, ..., p_{ik}\}$ and 0 otherwise, where p_{im} denotes the m-th *top-k* index. It can be seen that the expectations of each probability distribution are calibrated by the hyper-parameter δ to be close to the corresponding p_{im}. In this way, we can obtain sparse and differentiable one-hot vectors by adjusting the size of δ, as shown in Fig. 3(c).

3) Deformable Spatial Sampling: With the above derivation, we sample k neighbor joints with the highest similarity by multiplying the calibrated one-hot vectors. Specifically, the m-th neighbor joint of $\mathbf{x}_c$ is sampled as follows:

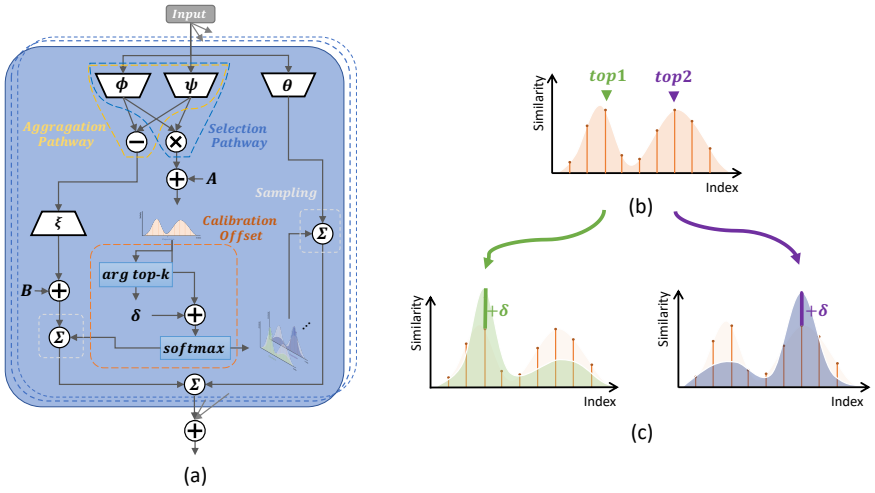


Fig. 3: Illustration of (a) the proposed DeSGC module, and an example of (b) the multimodal distribution and (c) the k well-calibrated probability distributions of corresponding indices. $\oplus$ and $\ominus$ denote element-wise summation and joint pair-wise subtraction, respectively. The dotted boxes represent different heads.

$$\tilde{\mathbf{x}}_c^{(m)} = \sum_{j=1}^{N} \theta(\mathbf{x}_j)\widehat{\pi}_{cj}^{(m)}, \tag{4}$$

where $\theta(\cdot)$ denotes a linear projection function for feature transformation, mapping channels from C_{in} to C_{out}, similar to the $\phi(\cdot)$ and $\psi(\cdot)$ in Equation (2). Equation (4) can be interpreted as that each neighbor joint is sampled by calculating the expectation according to their calibrated probability distribution.

4) Aggregation Pathway: Next, we construct a weight function to aggregate the sampled k neighbor joints. Specifically, we first calculate the weight matrix as in [22]:

$$w(\mathbf{x}_i, \mathbf{x}_j) = b_{ij} + \beta \cdot \xi(\sigma(\frac{1}{T}\sum_{t}^{T}(\phi(\mathbf{x}_i) - \psi(\mathbf{x}_j)))), \tag{5}$$

where b_{ij} denotes the learnable adjacency matrix component, β is a learnable scalar for balancing the intensity of sample-specific weight, $\xi(\cdot)$ denotes a linear transformation projected to C_{out} for distance-to-weight transformation. Note that $\phi(\cdot)$ and $\psi(\cdot)$ in Equation (5) are shared with Equation (2). We then extract the k weights corresponding to the sampled joints using calibrated one-hot vectors, similar to Equation (4). The weight of m-th neighbor joint at the center joint $\mathbf{x}_c$ is formulated as

$$\widehat{\mathbf{w}}_{c,m} = \sum_{j=1}^{N} w(\mathbf{x}_c, \mathbf{x}_j)\widehat{\pi}_{cj}^{(m)}. \tag{6}$$

Note that $\widehat{\mathbf{w}}_{c,m}$ is only for message aggregation and is distinguished from the selection pathway.

5) Spatial Graph Convolution: After defining the deformable sampling strategy and the weight function, we reformulate Equation (1), such that the proposed deformable spatial graph convolution can be formulated as

$$\mathbf{y}_c = \sum_{m=1}^{k} \widehat{\mathbf{w}}_{c,m}\tilde{\mathbf{x}}_c^{(m)}, \tag{7}$$

where $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N} \in R^{C_{out} \times T \times N}$. Note that we use three multi-head in parallel to extract diverse representations and fuse them via element-wise summation (See Fig. 3(a)).

6) Discussion: We provide a more in-depth analysis of DeSGC compared with vanilla GC as follows: (1) The entire process of our deformable graph convolution is differentiable through calibration offset. Thus, the receptive field can be adaptively learned in an end-to-end manner according to the pair-wise similarities for each action sample. Notably, different from the deformation vector generated by the input features as in [41], our calibration offset is a hyper-parameter that is used to approximate one-hot vectors, not changing the indices of key joints. (2) From Equation (7), it can be seen that only the k sampled joints participate in graph convolution, which becomes sparse and effective since the non-informative connections are eliminated. (3) Considering that non-informative joints may also exist in the k joints, we design the aggregation pathway for weighted message-passing separately from the selection pathway to further mitigate these irrelevant connections (See Section VI). Consequently, it becomes easier for us to accurately sample and aggregate the most informative joints.

B. Deformable Temporal GC (DeTGC)

While human action is essentially continuous, skeleton sequences are sampled discretely at a specific frame rate. This inevitably leads to a loss of information between frames, which gets worse as several pooling layers reduce the number of frames. However, the sampling locations of the vanilla temporal convolution (TC) [16] are set manually and discretely, which is not reasonable for modeling essentially continuous

action sequence. In this sub-section, we first describe how to perform deformable and continuous sampling on temporal graphs in a differentiable way, and then discuss its superiority over vanilla TC.

1) Deformable Temporal Sampling: We first interpret the sampling strategy of the vanilla TC at frame t_c as follows:

$$t_c^{(r)} = t_c + t_r, \tag{8}$$

where $t_r \in \mathcal{R}$ denotes the r-th relative sampling location, which is defined whthin the receptive field $\mathcal{R}$ determined by the kernel size. For example, if the kernel size is set to 3, $\mathcal{R}$ is $\{-1, 0, 1\}$. In this case, the sampling locations are discrete and fixed, which constrains the representation of complete continuous information. To obtain a continuous and dynamic receptive field, we first define the number of the locations η to sample, which is a hyper-parameter. We then uniformly sample η locations over the maximum range of $\mathcal{R}$, and set them as learnable parameters $\tilde{t}_r$. Here, the same configurations for the maximum range of $\mathcal{R}$ are set as the one suggested in [22]. By substituting the existing fixed t_r in Equation (8) with a new learnable $\tilde{t}_r$, we arrive at

$$\tilde{t}_c^{(r)} = t_c + \tilde{t}_r, \tag{9}$$

where $r = 1, ..., \eta$. In this case, since the learned index $\tilde{t}_c^{(r)}$ is a real number rather than a fixed integer, we extract the corresponding features via linear interpolation to obtain continuous data.

2) Temporal Graph Convolution: We follow the feature aggregation approach of the vanilla TC, *i.e.*, a single 1-D convolution with the kernel size η along the temporal dimension:

$$\mathbf{z}_c = \sum_{r=1}^{\eta} \mathbf{w}_r \mathcal{T}(\mathbf{Y}, \tilde{t}_c^{(r)}), \tag{10}$$

where $\mathcal{T}(\cdot, \cdot)$ denotes the sampling function via linear interpolation and $\mathbf{w}_r$ denotes the r-th element of the convolutional filter.

3) Discussion: Compared with vanilla TC, our proposed DeTGC improves in various areas: (1) In TC, the kernel size is a hyper-parameter that directly determines the size of the receptive field, whereas in our method, it is only a factor used to initialize the sampling locations. (2) The proposed DeTGC is a fully differentiable module and can be trained in an end-to-end manner, thus eventually obtaining dynamic and continuous receptive fields that can be suitable for different layers and datasets. In other words, we can get rid of the constraint of selecting the kernel size according to the input sequence length. (3) The continuous data mitigates the information loss caused by pooling. These demonstrate that our DeTGC has a more robust representation capability and flexibility.

V. Experiments

A. Datasets

We conduct extensive experiments on three widely used large-scale skeleton-based action recognition datasets, NTU RGB+D [26], NTU RGB+D 120 [27], and NW-UCLA [28], for a fair comparison with SOTA methods [22], [31].

1) NTU RGB+D: [26] is a large-scale human action recognition dataset containing 56,880 sequences over 60 classes. It provides the 3D Cartesian coordinates of 25 joints, which are captured from 3 Microsoft Kinect v2 cameras with different viewpoints, for each human in an action sample. Each action sample is performed by 40 volunteers in different age groups. The authors recommend two evaluation benchmarks: (1) Cross-Subject (X-Sub), where the 40 subjects are divided into training and testing groups. (2) Cross-View (X-View), where the data from camera views 2 and 3 are used for training, and data from camera view 1 is used for testing.

2) NTU RGB+D 120: [27] is an extended version of NTU RGB+D with an additional 60 action classes, with a total of 113,945 sequences. Similarly, the authors recommend two evaluation benchmarks: (1) Cross-Subject (X-Sub), where the 106 subjects are divided into training and testing groups. (2) Cross-Setup (X-Set), where the data from samples with even setup IDs are used for training, and data from samples with odd setup IDs are used for testing.

3) NW-ULCA: [28] is a human action recognition dataset containing 1,494 sequences over 10 classes captured from 3 Kinect cameras. Following the evaluation protocol from [28], we use the viewpoints of the first two cameras for training and the other for testing.

B. Implementation Details

Our extensive experiments are implemented with an RTX3090 GPU using the PyTorch framework. We set the calibration offset δ to 10, and the number of sampling joints k and η to 8 and 4, respectively. During the training, the stochastic gradient descent (SGD) with momentum 0.9 and weight decay 0.0004 is applied in the optimization. The training epoch is set to 80, and a warmup strategy [45] is used in the first 5 epochs to make training more stable. The initial learning rate is set to 0.1 and decays with a cosine schedule [46]. In addition, label smoothing [47] with a weight of 0.1 is adopted. Each sample is resized to 64 frames by linear interpolation as in [22]. The activation function is chosen as the LeakyReLU [48] function.

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

In this sub-section, we compare the proposed DeGCN with other state-of-art methods on three datasets. It is worth noting that, with the development of GCN-based methods, each of the various algorithms adopted different data transformation modules, which resize the original skeleton sequence to different lengths, we thus only provide the recognition accuracies and numbers of parameters except FLOPs for fair comparisons. Table I and Table II show the recognition performance of our DeGCN exceeds existing methods for all benchmarks.[1] Compared with the ST-GCN [16] in table I, which is the first GCN-based method and is currently the most popular backbone for skeleton-based action recognition, our model outperforms by around 10% and 20% in accuracy on both

[1]For benchmarks that lack reporting in table I, our experiments based on their public codes are presented.

TABLE I: Top-1 Accuracy(%) and Parameter Comparison with State-of-the-art Methods on NTU RGB+D and NTU RGB+D 120 Datasets. The Upper Block of Methods is for Non-graph-based Methods. "-" Denotes the Experimental Results Not Provided in the Reference, and "*" Indicates the Result Based on Using Public Codes. Bold and Underlined Figures Indicate the Best and Second Best Results for Each Dataset, Respectively.

Method	Conference	NTU RGB+D		NTU RGB+D120		#params (M)
		X-Sub	X-View	X-Sub	X-Set	
ST-LSTM [49]	ECCV16	69.2	77.7	-	-	-
GCA-LSTM [50]	CVPR17	74.4	82.8	-	-	-
Ta-CNN+ (4-ensemble) [51]	AAAI22	90.7	95.1	85.7	87.3	4.24
ST-GCN [16]	ECCV18	81.5	88.3	71.7*	72.2*	3.10*
AS-GCN [52]	CVPR19	86.8	94.2	78.2*	77.7*	9.50*
2s-AGCN (2-ensemble) [18]	CVPR19	88.5	95.1	82.7*	84.5*	6.94*
MS-AAGCN (4-ensemble) [29]	TIP20	90.0	96.2	-	-	15.12
Shift-GCN (4-ensemble) [53]	CVPR20	90.7	96.5	85.9	87.6	2.76*
MS-G3D (2-ensemble) [20]	CVPR20	91.5	96.2	86.9	88.4	6.40
DynamicGCN (4-ensemble) [23]	ACMMM20	91.5	96.0	87.3	88.6	-
Shift-GCN++ (4-ensemble) [54]	TIP21	90.5	96.3	85.6	87.2	1.80
MST-GCN (4-ensemble) [55]	AAAI21	91.5	96.6	87.5	88.8	12.00
AdaSGN (3-ensemble) [32]	ICCV21	90.5	95.3	85.9	86.8	5.36*
CTR-GCN (4-ensemble) [22]	ICCV21	92.4	96.8	88.9	90.6	5.84
FGCN (2-ensemble) [56]	TIP22	90.2	96.3	85.4	87.4	-
STF (2-ensemble) [25]	AAAI22	92.5	96.7	88.9	89.9	-
InfoGCN (4-ensemble) [31]	CVPR22	92.7	96.9	89.4	90.7	6.28
InfoGCN (6-ensemble) [31]	CVPR22	<u>93.0</u>	<u>97.1</u>	<u>89.8</u>	<u>91.2</u>	9.42
ML-STGNet (2-ensemble) [57]	TIP23	91.9	96.2	88.6	90.0	5.76
DeGCN (4-ensemble, Ours)	-	**93.6**	**97.4**	**91.0**	**92.1**	5.56

TABLE II: Top-1 Accuracy Comparison with State-of-the-art Methods on Northwestern-UCLA Dataset. Bold and Underlined Figures Indicate the Best and Second Best Results, Respectively.

Method	Acc.(%)
Lie Group [58]	74.2
Actionlet ensemble [59]	76.0
HBRNN-L [6]	78.5
Ensemble TS-LSTM [60]	89.2
AGC-LSTM (2-ensemble) [52]	93.3
Shift-GCN (4-ensemble) [53]	94.6
DC-GCN+ADG (4-ensemble) [61]	95.3
CTR-GCN (4-ensemble) [22]	96.5
FGCN (2-ensemble) [56]	95.3
InfoGCN (4-ensemble) [31]	96.6
InfoGCN (6-ensemble) [31]	<u>97.0</u>
DeGCN (4-ensemble, Ours)	**97.2**

benchmarks of both datasets, respectively. Compared with the current state-of-the-art method InfoGCN [31], our model improves performance by 0.9% and 1.6% over InfoGCN (4-ensemble) on X-sub benchmarks of NTU-RGB+D and NTU-RGB+D 120, respectively, with similar model size. Even, our model is 0.6% and 1.2% higher than InfoGCN with 6-ensembles, which has 1.69× more parameters than ours. These improvements are significantly higher than the increase rates of InfoGCN (0.3% and 0.5%, respectively) with the same number of ensembles compared to the previous SOTA method CTR-GCN [22]. These results are also keeping pace in the typical 3D action recognition dataset Northwestern-UCLA. As Shown in Table II, the proposed DeGCN again achieves the best accuracy of 97.2%, surpassing the previous state-of-the-art methods.

D. Ablation Studies

In this sub-section, we employ a model that replaces our basic block with an conventional ST-GC [16] module as a baseline. Ablation experiments are carried out for different network settings and analyzed with cross-subject (X-sub) benchmark on NTU RGB+D 120 using the joint modality.

1) Deformable Graph Convolution: To verify the effectiveness of the proposed deformable graph convolution, we compare our method with different methods, including: (1) local GC, implemented by the baseline [16], (2) global GC, which aggregates all joints as a weights function in Equation (6) for spatial modeling and as implemented in [43] for temporal modeling, and (3) the proposed deformable GC, which corresponds to the DeSGC module for spatial modeling and the DeTGC module for temporal modeling. It can be seen from Table III, among all the compared strategies, our deformable graph convolutions significantly improve performance for both spatial and temporal modeling. In particular, our deformable GC outperforms global GC by eliminating non-informative joints, underscoring the superiority of the separated selection pathway over relying solely on the aggregation pathway (*i.e.*, weight function). These results validate the effectiveness of our approach, which improves performance by adaptively conducting deformable sampling.

2) Component Studies: We conduct ablation experiments on the contributions of each DeGCN component and analyze them in terms of accuracy and model size. The results are shown in Table IV, from which it can be seen that the performance gradually improves as more components are used. Specifically, when configuring spatial-temporal modeling modules with the proposed DeSGC and DeTGC, our model improves performance by 2.2% over the baseline. Then, applying TSM reduces the parameters by 2.1 times without degrading the performance. This clearly verifies that it is efficient to learn spatial representations separately for each

TABLE III: Top-1 Accuracy Comparison with Different GC Methods in Spatial and Temporal Modeling on NTU RGB+D 120 X-Sub

Method	Acc.(%)	
	Spatial	Temporal
Local GC	83.8	83.8
Global GC	85.1	83.6
Deformable GC	**85.5**	**84.4**

TABLE IV: Top-1 Accuracy Comparison with Different Components on NTU RGB+D 120 X-Sub. "+" Denotes Maintaining the Current Setting and Adding More

Method	FLOPs	Param.	Acc.(%)
Baseline	1.65G	1.21M	83.8
+ DeSGC	1.81G	1.45M	$85.5^{\uparrow 1.7}$
+ DeTGC	1.78G	1.42M	$86.0^{\uparrow 2.2}$
+ TSM	**0.86G**	**0.69M**	$86.1^{\uparrow 2.3}$
+ 2-Branch	1.72G	1.39M	$\mathbf{87.6}^{\uparrow 3.8}$

TABLE V: Top-1 Accuracy Comparison with Multi-Modal Ensemble on NTU RGB+D 120 X-Sub

Method	FLOPs	Param.	Acc.(%)
Baseline (4-ensemble)	6.60G	4.84M	89.3
Joint	1.72G	1.39M	87.6
Bone	1.72G	1.39M	88.5
Velocity	1.72G	1.39M	83.7
JBF	1.72G	1.39M	89.6
Joint+Bone	3.44G	2.78M	$89.9^{\uparrow 0.6}$
Joint+Bone (3-branch)	5.16G	4.17M	$90.1^{\uparrow 0.8}$
Joint+Bone+JBF	5.16G	4.17M	$90.7^{\uparrow 1.4}$
Joint+Bone+Velocity+JBF	6.88G	5.56M	$\mathbf{91.0}^{\uparrow 1.7}$

temporal scale. Finally, for a fair comparison, we adopt a two-branch structure with a similar model size to be the baseline. Our model ultimately results in a 3.8% performance improvement.

3) Multi-Modal Ensemble with JBF: We evaluated each modality and their weighted fusion results, as shown in Table V. After combining the joint modality with the bone modality, the performance of the model can be improved from 87.6% to 89.9%. Furthermore, the performance reaches 90.7% when the JBF stream is applied to the ensemble. In contrast, without the JBF stream, our three-branch DeGCN increases by only 0.2% to 90.1% with the same parameters. This shows that the proposed JBF stream can significantly elevate the effect of ensemble by learning the correlation between the joint and bone modalities. Finally, Our full model reaches 91.0% in accuracy and outperforms the baseline model by 1.7% with the same ensemble setup.

4) Hyper-parameters in DeGCN: We first evaluate the proposed DeSGC module under various settings, including two hyper-parameters δ and k, on NTU RGB+D 120 X-Sub. The results, shown in Table VI and Fig. 4, demonstrate that our model is not sensitive to hyper-parameter settings around the best result with $\delta = 10$ and $k = 8$. We thus adopt the same configuration on all the benchmarks. It's worth mentioning, as shown in Fig. 4, both too small and too large values of k result in gradually degraded recognition performance. This verifies that our model is effective to select informative joints for improving action recognition. Also, if δ is too small,

TABLE VI: Top-1 Accuracy Comparison of the Proposed DeSGC Modules with Different Values of δ on NTU RGB+D 120 X-Sub

δ	0	5	10	15	20
Acc. (%)	84.4	85.3	**85.5**	85.2	84.9

TABLE VII: Top-1 Accuracy Comparison of the Proposed DeTGC Modules with Different Values of η on NTU RGB+D 120 X-Sub

η	2	3	4	5
Acc. (%)	84.0	84.2	**84.4**	84.3

TABLE VIII: Top-1 Accuracy (%) Comparison of the Proposed JBF and Multi-Modal Ensemble with Different Values of L_1 on NTU RGB+D 120 X-Sub

L_1	0	1	2	3	4	9
Acc. (%)	88.7	89.4	**89.6**	89.3	89.1	89.2

as in our analysis in Section IV-A, the model degrades its performance due to the difference between the expectation and the real index. On the other hand, if δ is too large, it will approach the hard one-hot vector, which also degrades its performance due to the difficulty of gradient backpropagation. These results validate the effectiveness of our DeSGC module, which improves performance by conducting differentiable key joint selections.

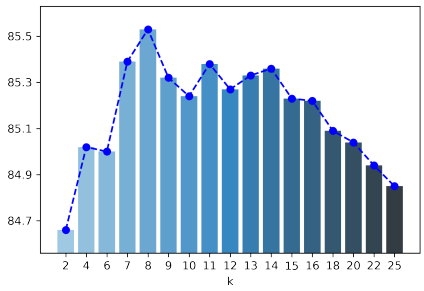


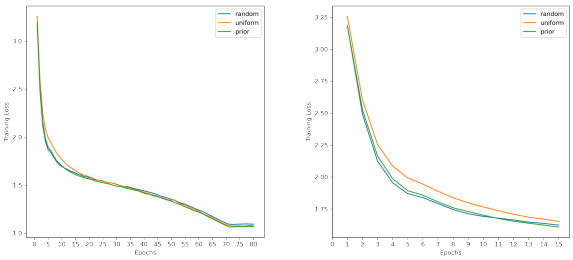
Fig. 4: Top-1 Accuracy Comparison of the Proposed DeSGC Modules with Different Values of k on NTU RGB+D 120 X-Sub.

Similary, we evaluate the proposed DeTGC module with various values of η. As shown in Table VII, the model is also not sensitive to hyper-parameter settings, and the best results can be obtained when η is an even number of 4. Considering the feature aggregation is performed by 1-D convolution corresponding to the number of sampling locations, as we describe in section IV-B, the results show that our DeTGC improves performance with more efficient model complexity compared to the baseline [16], which has a kernel size of 5.

Moreover, we evaluate the proposed JBF stream with various values of L_1. The recognition performances with different fusion stages are presented in Table VIII. The results demonstrate that the optimal performance is achieved when $L_1 = 2$ (*i.e.*, $L_2 = 7$), and it degrades with too small values of L_1,

TABLE IX: Top-1 Accuracy(%) Comparison of the Models with TSM and witouth TSM on NTU RGB+D 120 X-Sub. Numbers in Gray Denote the Results Reported in Their Papers. "*" Indicates the Result Based on Using Public Code

Method	FLOPs	Ratio	Param.	Ratio	Acc. (%)
ST-GCN* [16]	1.65G	-	1.21M	-	83.8
ST-GCN* + TSM	0.80G	↓ 2.1×	0.66M	↓ 1.8×	84.0
MS-G3D [20]	-	-	3.20M	-	-
MS-G3D* [20]	24.44G	-	3.20M	-	82.1
MS-G3D* + TSM	8.89G	↓ 2.7×	1.21M	↓ 2.6×	82.2
CTR-GCN [22]	1.97G	-	1.46M	-	84.9
CTR-GCN* [22]	1.97G	-	1.46M	-	85.1
CTR-GCN* + TSM	0.97G	↓ 2.0×	0.75M	↓ 1.9×	85.1
DeGCN (Ours)	1.78G	-	1.42M	-	86.0
DeGCN + TSM	0.86G	↓ 2.1×	0.69M	↓ 2.1×	86.1



(a) (b)

Fig. 5: Training loss curves for (a) the entire training process and (b) the first 15 epochs with different initialization strategy.

as the representation capabilities of each modality are limited. Conversely, too large values of L_1 also have relatively inferior performance due to the lack of correlation representations. Accordingly, as mentioned in Section III, we perform fusion of the joint and bone modalities after extracting discriminative representations through the stream consisting of an initial block and two basic blocks for each modality.

5) Initialization of k joints: Since the selected k joints can be changed during the training process, we compare three different initialization strategies to investigate their effectiveness, including random, uniform, and commonly used prior strategies. Notably, in our approach, the initialization for the k joints is equivalent to that of the similarity matrix. In this context, the prior strategy denotes initializing the adjacency matrix according to human natural connections, and both the random and uniform strategies also follow their distribution, respectively. As shown in Fig. 5, the training loss curve of uniform strategy exhibited slightly slower initial convergence compared to the prior or random strategies, as selecting the top-k values among the same similarities for all joints is challenging. Therefore, considering the stabilities both during the initial phase and throughout the entire training process, we adopt the prior strategy in our model.

E. TSM on Other GCs

Our proposed TSM can be seamlessly incorporated into current GCNs. In this sub-section, to verify the effectiveness of TSM, we apply our approach to other existing graph convolution methods utilizing multi-scale temporal modeling, including ST-GCN [16], MS-G3D [20], and CTR-GCN [22]. Our experimental results based on their public implementation code are presented in Table IX, including FLOPs, number of parameters, their reduction ratios, and accuracy. Note that ST-GCN in table IX is identical to the baseline adopted in Section V-D, $i.e.$, the model replacing our basic block with an conventional ST-GC module.

From the Table IX, it can be seen that the proposed TSM is beneficial for effectively reducing the model complexity (both FLOPs and number of parameters) without degrading its performance. This is especially true for MS-G3D, as they leverage several spatial graph convolution modules consisting of point-wise convolutions for each higher-order of the adjacency matrix to capture all distances. These results verify that it is efficient to learn spatial representations separately for each temporal scale.

F. Performance on Confusing Classes

We further analyze the accuracy difference (%) between our DeGCN and the baseline [16] for each action class on NTU RGB+D 120 X-Sub. As shown in Fig. 6, the highest differences tend to occur on the actions where two-hand correlations are most relevant, $e.g.$, the model offers improvement of 13.24% for "$writing$", 10.26% for "$ball\ up\ paper$", and 9.16% for "$reading$". These results demonstrate that our DeGCN has a robust performance for recognizing the actions that are very similar by focusing on the most informative joints.

For the lower performance of our model on a small subset of classes, we also provide a in-depth analysis as follows: (1) the average drop of these classes is 1.63%, which is much smaller than the increase rate of 9.96% for increased classes with the same number. (2) Our model reduces the variance of accuracy for all classes from 0.017 to 0.013. (3) We generate the confusion matrices for these classes that are normalized by the whole column ($i.e.$, predictions for 120 classes) to verify the precision of each class, as shown in Fig. 7. Compared with the baseline, our model improves the precision for almost all of these classes. (4) Considering the increase in mean accuracy and the reduction in variance for all classes, and the increase of precision indicate that we mitigate the overfitting of the baseline to easy classes where the accuracy is higher than the average.

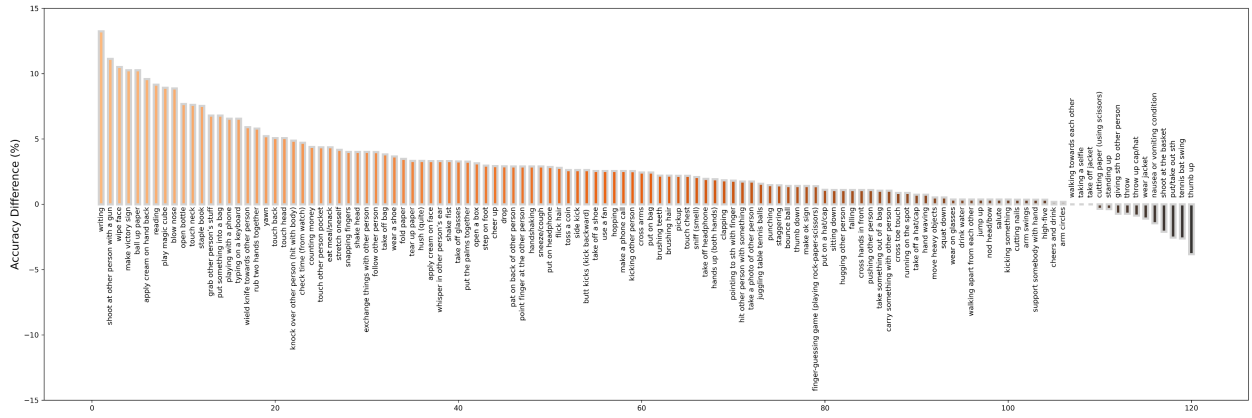
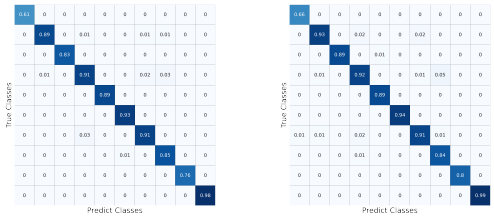


Fig. 6: Top-1 Accuracy Difference (%) between the proposed DeGCN and the baseline [16] with the joint input modality on NTU RGB+D 120 X-Sub.



(a) (b)

Fig. 7: Confusion matrices obtained by the (a) baseline and (b) DeGCN on NTU-RGB+D 120 X-Sub.

Furthermore, to validate the statistical significance of our improvements, we conduct a paired t-test for accuracy differences. The t-test is a commonly used statistical test that helps determine whether the observed difference between two sets of data is random or statistically significant. We set the significance level α as 0.05 for this experiments. From the results yielded, it was found that the confidence of the improvement is high (p-value $\ll 0.05$). Hence, we reject the null hypothesis that there is no difference in mean accuracy between the two models ($i.e.$, the baseline and our models, respectively), and state that we have significant evidence of our improvements.

VI. Visualizations of Deformable Kernels

A. *Kernels of DeSGC*

To better understand what the proposed DeGCN has learned internally, we visualize the learned deformable sampling locations. For the DeSGC module, we illustrate the key joint and the corresponding sampling locations of some actions, as shown in Fig. 8. We observe that our model successfully samples the most informative joints, $i.e.$, two arms for "$reading$", one arm for "$hand\ waving$", and the lower body for "$sitting\ down$". Especially for the action "$taking\ a\ selfie$", the head node adaptively samples the arm joints of holding the phone as neighbor joints. Moreover, Fig. 9 shows the sampling locations learned by different heads. It can be seen that our model can explore various informative joints through each head, just as the movements of the upper body lead to the $legs$ when "$jumping$".

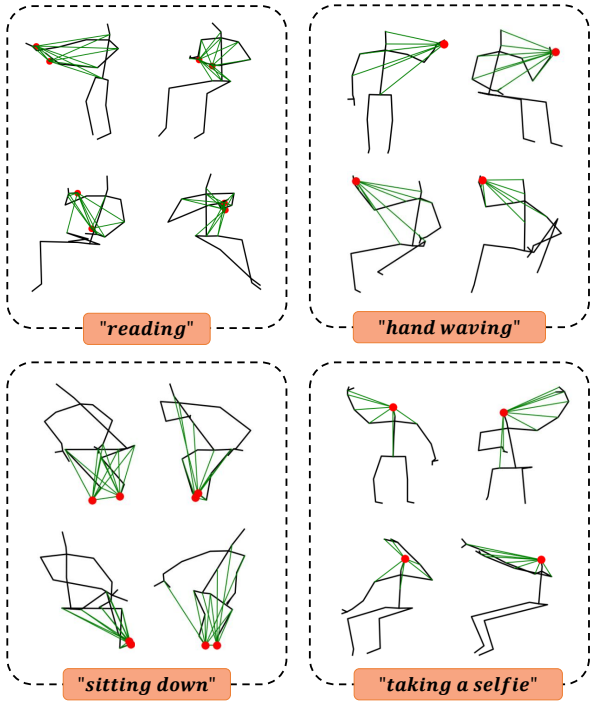


Fig. 8: Qualitative examples of the deformable sampling locations of the proposed DeSGC module on NTU RGB+D 120. The colored lines denote the sampling locations of the graph convolution. Our method adaptively samples the most informative joints regardless of intra-class variety.

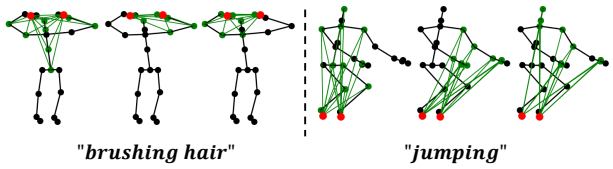


Fig. 9: Visualization of the sampling locations with multi-head. The three skeletons in each action represent the kernels from different heads.

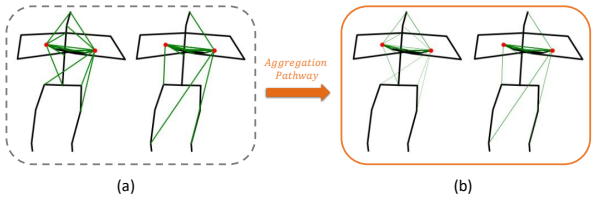


Fig. 10: Visualization of (a) the sampling locations and (b) the corresponding weights for action "*reading*". The thickness of the colored lines denotes the strength of the weight. Our method further mitigates non-informative connections via a separated aggregation pathway.

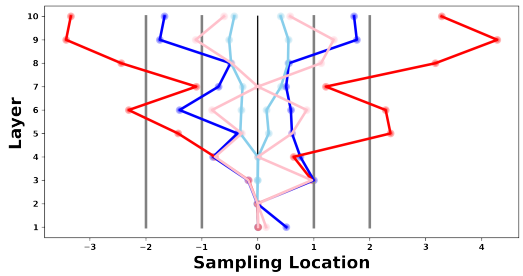
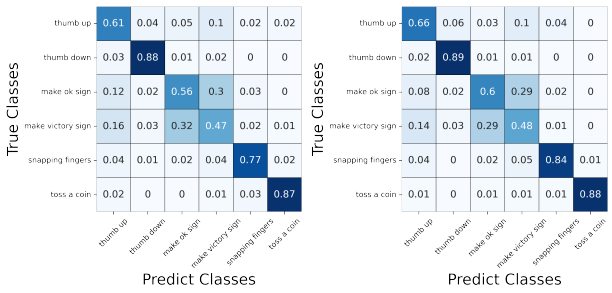


Fig. 11: Visualization of the deformable sampling locations of the proposed DeTGC module. Note that Blue and Skyblue, and Red and Pink, respectively, represent $\eta = 4$ locations of two different DeTGC modules. Gray represents the locations of vanilla TC with a kernel size is 5.

However, for some actions with a small number of key joints, such as "*reading*", the non-informative joints may also exist in the k sampled joints. Thus, as described in Section IV-A, we design the aggregation pathway for weighted message-passing separately from the sampling pathway to further mitigate negatively correlated connections. As shown in Fig. 10, the weights of non-informative joints (*i.e.*, legs) are mitigated. This is a consistent result of our motivation.

B. *Kernels of DeTGC*

For the DeTGC module, we provide zero-centered sampling locations from the bottom layer to the top layer, as shown in Fig. 11. We find that the receptive field of the bottom layers (corresponding y-axis 1 and 2) is close to zero, which means that the focus is on learning the initial topology via



(a) (b)

Fig. 12: Confusion matrices for "*thumb up*" and its Top-5 relevant classes obtained by the (a) baseline and (b) DeGCN on NTU-RGB+D 120 X-Sub.

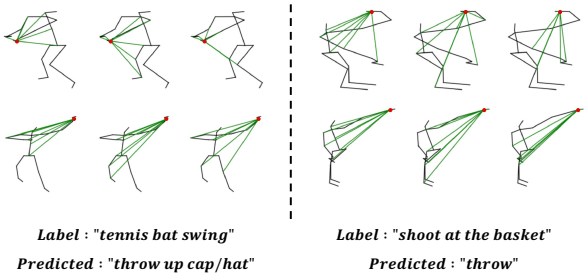


Fig. 13: Qualitative examples of the failure cases. The three skeletons in each action represent the kernels from different heads.

spatial modeling, and then the range of focus tends to expand gradually as it approaches the top layer. As such, compared with the vanilla TC, our model can access dynamic and continuous receptive fields from local to global, by which fluid and effective temporal information is captured.

C. *Failure Cases*

We conducted a comprehensive analysis, particularly with the baseline, and identified our failure cases in two categories: actions requiring two-hand correlations and those heavily reliant on global joints. In the former, as validated above, our model demonstrated relatively robust performance by eliminating non-informative joints. For instance, although the "*thumb up*" gesture showed the largest accuracy drop, our model exhibited improved precision across relevant classes, as illustrated in the normalized confusion matrices in Fig. 12. However, their accuracies still fall behind the average due to the use of crude and noisy pre-defined hand topologies, such as being limited to only 4 joints for NTU RGB+D. In the latter category, as our model can explore up to k joints for one kernel in multi-head mechanisms, its recognition occasionally faltered for actions requiring information from all joints, leading to confusion with locally relevant actions, such as *shoot at the basket* and *throw* (See Fig. 13). While this

issue is infrequent and less critical than the former, it remains a focus for future improvements.

VII. Conclusion

We present DeGCN, a novel framework for skeleton-based action recognition, which enables a sampling operation to be differentiable and deformable, empowering end-to-end learning of discriminative receptive fields for both spatial and temporal graphs. The results show that our DeSGC and DeTGC, adaptively exploring the most relevant joints for various human actions, outperforms both existing local and global graph convolution methods. Furthermore, we introduce TSM and JBF to achieve a better trade-off between accuracy and model size. TSM, which reduces redundant computation for modeling spatial relations, effectively refines conventional spatial-temporal modeling. The JBF stream can elevate the effect of ensemble between the joint and bone modalities. Our experiments on three benchmark datasets suggest that DeGCN achieves state-of-the-art performance. Finally, our deformable approach can be applied not only to skeleton-based action recognition, but also to various graph-related domains such as social network analysis and recommendation systems, which is an interesting direction for future work. In addition, adapting different k values for each joint would be effective in enhancing robustness across various graph scales. We will investigate various strategies, such as soft ranking or threshold-based learning, to achieve this goal.

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.
[2] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, no. 1, pp. 935–942, 2009.
[3] Y. G. Jiang, Q. Dai, W. Liu, X. Xue, and C. W. Ngo, "Human Action Recognition in Unconstrained Videos by Explicit Motion Modeling," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3781–3795, 2015.
[4] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 588–595, 2014.
[5] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014.
[6] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1110–1118, 2015.
[7] 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," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2136–2145, 2017.
[8] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream Recurrent Neural Networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3633–3642, 2017.
[9] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2017, pp. 597–600.
[10] T. S. Kim and A. Reiter, "Interpretable 3D Human Action Analysis with Temporal Convolutional Networks," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 1623–1631, 2017.
[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–14, 2017.
[12] A. Mazari and H. Sahbi, "Mlgcn: Multi-laplacian graph convolutional networks for human action recognition," in *The British Machine Vision Conference (BMVC)*, 2019.
[13] H. Sahbi, "Learning laplacians in chebyshev graph convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2064–2075.
[14] F. Monti, D. Boscaini, J. Masci, E. Rodoì, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," p. 5115–5124, 2017.
[15] H. Sahbi, "Learning connectivity with graph convolutional networks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9996–10 003.
[16] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 7444–7452, 2018.
[17] C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, "Spatio-temporal graph convolution for skeleton based action recognition," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 3482–3489, 2018.
[18] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 12 018–12 027, 2019.
[19] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 3590–3598, 2019.
[20] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 143–152.
[21] H. Sahbi, "Kernel-based graph convolutional networks." Institute of Electrical and Electronics Engineers Inc., 2020, pp. 4887–4894.
[22] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 13 339–13 348, 2021.
[23] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, and H. Tang, "Dynamic GCN: Context-enriched Topology Learning for Skeleton-based Action Recognition," *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, pp. 55–63, 2020.
[24] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1109–1118, 2020.
[25] L. Ke, K.-C. Peng, and S. Lyu, "Towards to-a-t spatio-temporal focus for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 1131–1139.
[26] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 1010–1019, 2016.
[27] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Y. Duan, and A. C. Kot, "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2684–2701, 2020.
[28] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. C. Zhu, "Cross-view action modeling, learning, and recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2649–2656, 2014.
[29] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.
[30] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Constructing stronger and faster baselines for skeleton-based action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 2, pp. 1474–1488, 2023.
[31] H.-g. Chi, M. H. Ha, S. Chi, S. W. Lee, Q. Huang, and K. Ramani, "Infogcn: Representation learning for human skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 186–20 196.

[32] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "AdaSGN: Adapting Joint Number and Model Size for Efficient Skeleton-Based Action Recognition," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 13 393–13 402, 2021.
[33] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The handbook of brain theory and neural networks*, 1995, pp. 255–258.
[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
[36] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 764–773, 2017.
[37] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021.
[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.
[39] Z. Chen, Y. Zhu, C. Zhao, G. Hu, W. Zeng, J. Wang, and M. Tang, "DPT: Deformable Patch-based Transformer for Visual Recognition," *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2899–2907, 2021.
[40] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," pp. 568–578, 2021.
[41] J. Park, S. Yoo, J. Park, and H. J. Kim, "Deformable graph convolutional networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7949–7956.
[42] H. Xia and X. Gao, "Multi-scale mixed dense graph convolution network for skeleton-based action recognition," *IEEE Access*, vol. 9, pp. 36 475–36 484, 2021.
[43] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Decoupled Spatial-Temporal Attention Network for Skeleton-Based Action-Gesture Recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12626 LNCS, pp. 38–53, 2021.
[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016.
[46] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–16, 2017.
[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016.
[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, vol. 28, 2013.
[49] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9907 LNCS, pp. 816–833, 2016.
[50] J. Liu, G. Wang, P. Hu, L. Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3671–3680, 2017.
[51] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware Convolutional Neural Network for Efficient Skeleton-based Action Recognition," vol. 36, no. 3, pp. 2866–2874, 2021.
[52] X. Ding, K. Yang, and W. Chen, "An attention-enhanced recurrent graph convolutional network for skeleton-based action recognition," *ACM International Conference Proceeding Series*, pp. 79–84, 2019.
[53] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 180–189, 2020.
[54] K. Cheng, Y. Zhang, X. He, J. Cheng, and H. Lu, "Extremely lightweight skeleton-based action recognition with shiftgcn++," *IEEE Transactions on Image Processing*, vol. 30, pp. 7333–7348, 2021.
[55] K. Yang, X. Ding, and W. Chen, "Multi-scale spatial temporal graph convolutional LSTM network for skeleton-based human action recognition," *ACM International Conference Proceeding Series*, pp. 3–9, 2019.
[56] H. Yang, D. Yan, L. Zhang, Y. Sun, D. Li, and S. J. Maybank, "Feedback graph convolutional network for skeleton-based action recognition," *IEEE Transactions on Image Processing*, vol. 31, pp. 164–175, 2021.
[57] Y. Zhu, H. Shuai, G. Liu, and Q. Liu, "Multilevel spatial-temporal excited graph network for skeleton-based action recognition," *IEEE Transactions on Image Processing*, vol. 32, pp. 496–508, 2023.
[58] V. Veeriah, N. Zhuang, and G. J. Qi, "Differential recurrent neural networks for action recognition," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 4041–4049, 2015.
[59] J. Wang, Z. Liu, and Y. Wu, "Learning Actionlet Ensemble for 3D Human Action Recognition," *SpringerBriefs in Computer Science*, vol. 0, no. 9783319045603, pp. 11–40, 2014.
[60] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 1012–1020, 2017.
[61] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling GCN with DropGraph Module for Skeleton-Based Action Recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12369 LNCS, pp. 536–553, 2020.