Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition

Ziyu Liu[1,3], Hongwen Zhang[2], Zhenghao Chen[1], Zhiyong Wang[1], Wanli Ouyang[1,3]
[1]The University of Sydney   [2]University of Chinese Academy of Sciences & CASIA
[3]The University of Sydney, SenseTime Computer Vision Research Group, Australia

Abstract

Spatial-temporal graphs have been widely used by skeleton-based action recognition algorithms to model human action dynamics. To capture robust movement patterns from these graphs, long-range and multi-scale context aggregation and spatial-temporal dependency modeling are critical aspects of a powerful feature extractor. However, existing methods have limitations in achieving (1) unbiased long-range joint relationship modeling under multi-scale operators and (2) unobstructed cross-spacetime information flow for capturing complex spatial-temporal dependencies. In this work, we present (1) a simple method to disentangle multi-scale graph convolutions and (2) a unified spatial-temporal graph convolutional operator named G3D. The proposed multi-scale aggregation scheme disentangles the importance of nodes in different neighborhoods for effective long-range modeling. The proposed G3D module leverages dense cross-spacetime edges as skip connections for direct information propagation across the spatial-temporal graph. By coupling these proposals, we develop a powerful feature extractor named MS-G3D based on which our model[1] outperforms previous state-of-the-art methods on three large-scale datasets: NTU RGB+D 60, NTU RGB+D 120, and Kinetics Skeleton 400.

1. Introduction

Human action recognition is an important task with many real-world applications. In particular, *skeleton-based* human action recognition involves predicting actions from skeleton representations of human bodies instead of raw RGB videos, and the significant results seen in recent work [50, 33, 32, 34, 21, 20, 54, 35] have proven its merits. In contrast to RGB representations, skeleton data contain only the 2D [50, 15] or 3D [31, 25] positions of the human key joints, providing highly abstract information that is also free of environmental noises (*e.g.* background clutter, light-
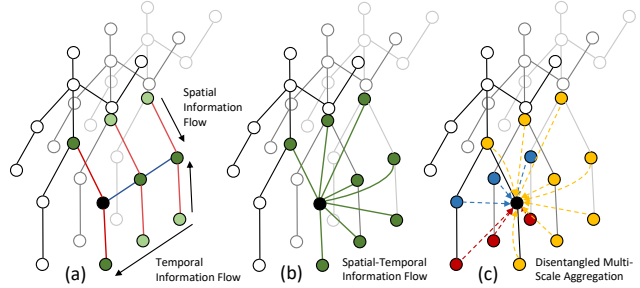

Figure 1: (a) Factorized spatial and temporal modeling on skeleton graph sequences causes *indirect* information flow. (b) In this work, we propose to capture cross-spacetime correlations with *unified* spatial-temporal graph convolutions. (c) Disentangling node features at separate spatial-temporal neighborhoods (yellow, blue, red at different distances, partially colored for clarity) is pivotal for effective multi-scale learning in the spatial-temporal domain.

ing conditions, clothing), allowing action recognition algorithms to focus on the robust features of the action.

Earlier approaches to skeleton-based action recognition treat human joints as a set of independent features, and they model the spatial and temporal joint correlations through hand-crafted [42, 43] or learned [31, 6, 48, 54] aggregations of these features. However, these methods overlook the inherent relationships between the human joints, which are best captured with human skeleton graphs with joints as nodes and their natural connectivity (*i.e.* "bones") as edges. For this reason, recent approaches [50, 19, 34, 35, 32] model the joint movement patterns of an action with a skeleton *spatial-temporal graph*, which is a series of disjoint and isomorphic skeleton graphs at different time steps carrying information in both spatial and temporal dimensions.

For robust action recognition from skeleton graphs, an ideal algorithm should look beyond the local joint connectivity and extract multi-scale structural features and long-range dependencies, since joints that are structurally apart can also have strong correlations. Many existing approaches achieve this by performing *graph convolutions* [17] with higher-order polynomials of the skeleton adjacency matrix: intuitively, a powered adjacency matrix cap-

---

[1]Code is available at github.com/kenziyuliu/ms-g3d

tures the number of walks between every pair of nodes with the length of the walks being the same as the power; the adjacency polynomial thus increases the receptive field of graph convolutions by making distant neighbors reachable. However, this formulation suffers from the *biased weighting problem*, where the existence of *cyclic* walks on undirected graphs means that edge weights will be biased towards closer nodes against further nodes. On skeleton graphs, this means that a higher polynomial order is only marginally effective at capturing information from distant joints, since the aggregated features will be dominated by the joints from local body parts. This is a critical drawback limiting the scalability of existing multi-scale aggregators.

Another desirable characteristic of robust algorithms is the ability to leverage the complex cross-spacetime joint relationships for action recognition. However, to this end, most existing approaches [50, 33, 19, 32, 21, 34, 18] deploy interleaving spatial-only and temporal-only modules (Fig. 1(a)), analogous to factorized 3D convolutions [30, 39]. A typical approach is to first use graph convolutions to extract spatial relationships at each time step, and then use recurrent [19, 34, 18] or 1D convolutional [50, 33, 21, 32] layers to model temporal dynamics. While such factorization allows efficient long-range modeling, it hinders the direct information flow across spacetime for capturing complex regional spatial-temporal joint dependencies. For example, the action "standing up" often has co-occurring movements of upper and lower body across both space and time, where upper body movements (leaning forward) strongly correlate to the lower body's *future* movements (standing up). These strong cues for making predictions may be ineffectively captured by factorized modeling.

In this work, we address the above limitations from two aspects. First, we propose a new multi-scale aggregation scheme that tackles the biased weighting problem by removing redundant dependencies between further and closer neighborhoods, thus disentangling their features under multi-scale aggregation (illustrated in Fig. 2). This leads to more powerful multi-scale operators that can model relationships of joints irrespective of the distances between them. Second, we propose G3D, a novel unified spatial-temporal graph convolution module that *directly* models cross-spacetime joint dependencies. G3D does so by introducing graph edges across the "3D" spatial-temporal domain as skip connections for unobstructed information flow (Fig. 1(b)), substantially facilitating spatial-temporal feature learning. Remarkably, our proposed disentangled aggregation scheme augments G3D with multi-scale reasoning in spacetime (Fig. 1(c)) without being affected by the biased weighting problem, despite extra edges were introduced. The resulting powerful feature extractor, named MS-G3D, forms a building block of our final model architecture that outperforms state-of-the-art methods on three large-

scale skeleton action datasets: NTU RGB+D 120 [25], NTU RGB+D 60 [31], and Kinetics Skeleton 400 [15]. The main contributions of this work are summarized as follows:

(i) We propose a disentangled multi-scale aggregation scheme that removes redundant dependencies between node features from different neighborhoods, which allows powerful multi-scale aggregators to effectively capture graph-wide joint relationships on human skeletons.

(ii) We propose a unified spatial-temporal graph convolution (G3D) operator which facilitates direct information flow across spacetime for effective feature learning.

(iii) Integrating the disentangled aggregation scheme with G3D gives a powerful feature extractor (MS-G3D) with multi-scale receptive fields across both spatial and temporal dimensions. The direct multi-scale aggregation of features in spacetime further boosts model performance.

2. Related Work

2.1. Neural Nets on Graphs

Architectures. To extract features from arbitrarily structured graphs, Graph Neural Networks (GNNs) have been developed and explored extensively [5, 17, 3, 2, 10, 40, 49, 1, 7, 11, 22]. Recently proposed GNNs can broadly be classified into spectral GNNs [3, 11, 22, 13, 17] and spatial GNNs [17, 49, 10, 51, 41, 1, 45]. Spectral GNNs convolve the input graph signals with a set of learned filters in the graph Fourier domain. They are however limited in terms of computational efficiency and generalizability to new graphs due to the requirement of eigendecomposition and the assumption of fixed adjacency. Spatial GNNs, in contrast, generally perform layer-wise update for each node by (1) selecting neighbors with a neighborhood function (*e.g.* adjacent nodes); (2) merging the features from the selected neighbors and itself with an aggregation function (*e.g.* mean pooling); and (3) applying an activated transformation to the merged features (*e.g.* MLP [49]). Among different GNN variants, the Graph Convolutional Network (GCN) [17] was first introduced as a first-order approximation for localized spectral convolutions, but its simplicity as a mean neighborhood aggregator [49, 46] has quickly led many subsequent spatial GNN architectures [49, 1, 45, 7] and various applications involving graph structured data [44, 47, 52, 50, 33, 34, 21] to treat it as a *spatial* GNN baseline. This work adapts the layer-wise update rule in GCN.

Multi-Scale Graph Convolutions. Multi-scale spatial GNNs have also been proposed to capture features from non-local neighbors. [1, 19, 21, 45, 24] use higher order polynomials of the graph adjacency matrix to aggregate features from long-range neighbor nodes. Truncated Block Krylov network [29] similarly raises the adjacency matrix to

higher powers and obtains multi-scale information through dense features concatenation from different hidden layers. LanczosNet [24] deploys a low-rank approximation of the adjacency matrix to speed up the exponentiation on large graphs. As mentioned in Section 1, we argue that adjacency powering can have adverse effects on long-range modeling due to weighting bias, and our proposed module aims to address this with disentangled multi-scale aggregators.

2.2. Skeleton-Based Action Recognition

Earlier approaches [42, 6, 31, 36, 43, 48, 54] to skeleton-based action recognition focus on hand-crafting features and joint relationships for downstream classifiers, which ignore the important semantic connectivity of the human body. By constructing spatial-temporal graphs and modeling the spatial relationships with GNNs directly, recent approaches [50, 19, 8, 21, 8, 33, 32, 34, 18] have seen significant performance boost, indicating the necessity of the semantic human skeleton for action predictions.

An early application of graph convolutions is ST-GCN [50], where spatial graph convolutions along with interleaving temporal convolutions are used for spatial-temporal modeling. A concurrent work by Li et al. [19] presents a similar approach, but it notably introduces a multi-scale module by raising skeleton adjacency to higher powers. AS-GCN [21] also uses adjacency powering for multi-scale modeling, but it additionally generates human poses to augment the spatial graph convolution. Spatial-Temporal Graph Routing (STGR) network [18] adds extra edges to the skeleton graph using frame-wise attention and global self-attention mechanisms. Similarly, 2s-AGCN [33] introduces graph adaptiveness with self-attention along with a freely learned graph residual mask. It also uses a two-stream ensemble with skeleton bone features to boost performance. DGNN [32] likewise leverages bone features, but it instead simultaneously updates the joint and bone features through an alternating spatial aggregation scheme. Note that these approaches primarily focus on spatial modeling; in contrast, we present a unified approach for capturing complex joint correlations directly across spacetime.

Another relevant work is GR-GCN [8], which merges every three frames over the skeleton graph sequence and adds sparsified edges between adjacent frames. Whereas GR-GCN also deploys cross-spacetime edges, our G3D module has several important distinctions: (1) Cross-spacetime edges in G3D follow the semantic human skeleton, which is naturally a more interpretable and more robust representation than the sparsified, one-size-fits-all graph in GR-GCN. The underlying graph is also much easier to compute. (2) GR-GCN has cross-spacetime edges only between adjacent frames, which prevents it to reason beyond a limited temporal context of three frames. (3) G3D can learn from multiple temporal contexts simultaneously leveraging different win-

dow sizes and dilations, which is not addressed in GR-GCN.

3. MS-G3D

3.1. Preliminaries

Notations. A human skeleton graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_N\}$ is the set of N nodes representing joints, and $\mathcal{E}$ is the edge set representing bones captured by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ where initially $\mathbf{A}_{i,j} = 1$ if an edge directs from v_i to v_j and 0 otherwise. $\mathbf{A}$ is symmetric since $\mathcal{G}$ is undirected. Actions as *graph sequences* have a node features set $\mathcal{X} = \{x_{t,n} \in \mathbb{R}^C \mid t, n \in \mathbb{Z}, 1 \leq t \leq T, 1 \leq n \leq N\}$ represented as a feature tensor $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$, where $x_{t,n} = \mathbf{X}_{t,n,:}$ is the C dimensional feature vector for node v_n at time t over a total of T frames. The input action is thus adequately described by $\mathbf{A}$ structurally and by $\mathbf{X}$ feature-wise, with $\mathbf{X}_t \in \mathbb{R}^{N \times C}$ being the node features at time t. $\Theta^{(l)} \in \mathbb{R}^{C_l \times C_{l+1}}$ denotes a learnable weight matrix at layer l of a network.

Graph Convolutional Nets (GCNs). On skeleton inputs defined by features $\mathbf{X}$ and graph structure $\mathbf{A}$, the layer-wise update rule of GCNs can be applied to features at time t as:

$$\mathbf{X}_t^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}_t^{(l)}\Theta^{(l)}\right), \qquad (1)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the skeleton graph with added *self-loops* to keep identity features, $\tilde{\mathbf{D}}$ is the diagonal degree matrix of $\tilde{\mathbf{A}}$, and $\sigma(\cdot)$ is an activation function. The term $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}_t^{(l)}$ can be intuitively interpreted as an approximate spatial *mean* feature aggregation from the direct neighborhood followed by an activated linear layer.

3.2. Disentangled Multi-Scale Aggregation

Biased Weighting Problem. Under the spatial aggregation framework in Eq. 1, existing approaches [21] employ *higher-order polynomials* of the adjacency matrix to aggregate multi-scale structural information at time t, as:

$$\mathbf{X}_t^{(l+1)} = \sigma\left(\sum_{k=0}^{K}\widehat{\mathbf{A}}^k\mathbf{X}_t^{(l)}\Theta_{(k)}^{(l)}\right), \qquad (2)$$

where K controls the number of scales to aggregate. Here, $\widehat{\mathbf{A}}$ is a normalized form of $\mathbf{A}$, *e.g.* [19] uses the symmetric normalized graph Laplacian $\widehat{\mathbf{A}} = \mathbf{L}^{\text{norm}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$; [21] uses the random-walk normalized adjacency $\widehat{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$; more generally, one can use $\widehat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ from GCNs. It is easy to see that $\mathbf{A}_{i,j}^k = \mathbf{A}_{j,i}^k$ gives the number of length k walks between v_i and v_j, and thus the term $\widehat{\mathbf{A}}^k\mathbf{X}_t^{(l)}$ is performing a *weighted* feature average based on the number of such walks. However, it is clear that there are drastically more possible length k walks to closer nodes than to the actual k-hop neighbors due to cyclic
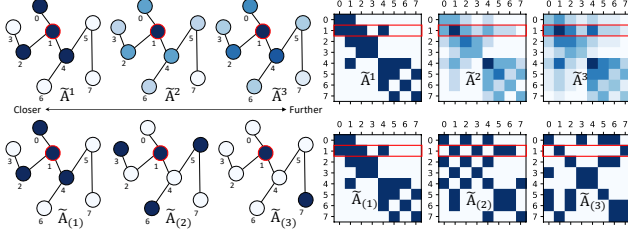
Figure 2: Illustration of the **biased weighting problem** and the proposed **disentangled aggregation scheme**. Darker color indicates higher weighting to the central node (red). **Top left**: closer nodes receive higher weighting from adjacency powering, which makes long-range modeling less effective, especially when multiple scales are aggregated. **Bottom left**: our proposed disentangled aggregation models joint relationships at each neighborhood while keeping identity features. **Right**: Visualizing the corresponding adjacency matrices. Node self-loops are omitted for visual clarity.

walks. This causes a bias towards the local region as well as nodes with higher degrees. The node self-loops in GCNs allow even more possible cycles (as walks can always cycle on self-loops) and thus amplify the bias. See Fig. 2 for illustration. Under multi-scale aggregation on skeleton graphs, the aggregated features will thus be dominated by signals from local body parts, making it ineffective to capture long-range joint dependencies with higher polynomial orders.

Disentangling Neighborhoods. To address the above problem, we first define the k-adjacency matrix $\tilde{\mathbf{A}}_{(k)}$ as

$$[\tilde{\mathbf{A}}_{(k)}]_{i,j} = \begin{cases} 1 & \text{if } d(v_i, v_j) = k, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $d(v_i, v_j)$ gives the shortest distance in number of hops between v_i and v_j. $\tilde{\mathbf{A}}_{(k)}$ is thus a generalization of $\tilde{\mathbf{A}}$ to further neighborhoods, with $\tilde{\mathbf{A}}_{(1)} = \tilde{\mathbf{A}}$ and $\tilde{\mathbf{A}}_{(0)} = \mathbf{I}$. Under spatial aggregation in Eq. 1, the inclusion of self-loops in $\tilde{\mathbf{A}}_{(k)}$ is critical for learning the *relationships* between the current joint and its k-hop neighbors, as well as for keeping each joint's *identity* information when no k-hop neighbors are available. Given that N is small, $\tilde{\mathbf{A}}_{(k)}$ can be easily computed, *e.g.*, using *differences* of graph powers as $\tilde{\mathbf{A}}_{(k)} = \mathbf{I} + \mathbb{1}\left(\tilde{\mathbf{A}}^k \geq 1\right) - \mathbb{1}\left(\tilde{\mathbf{A}}^{k-1} \geq 1\right)$. Substituting $\hat{\mathbf{A}}^k$ with $\tilde{\mathbf{A}}_{(k)}$ in Eq. 2, we arrive at:

$$\mathbf{X}_t^{(l+1)} = \sigma\left(\sum_{k=0}^K \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \tilde{\mathbf{A}}_{(k)} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \mathbf{X}_t^{(l)} \Theta_{(k)}^{(l)}\right), \quad (4)$$

where $\tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \tilde{\mathbf{A}}_{(k)} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}}$ is the normalized [17] k-adjacency.

Unlike the previous case where possible length k walks are predominantly conditioned on length $k-1$ walks, the proposed disentangled formulation in Eq. 4 addresses

the biased weighting problem by removing redundant dependencies of distant neighborhoods' weighting on closer neighborhoods. Additional scales with larger k are therefore aggregated in an *additive* manner under a multi-scale operator, making long-range modeling with large values of k to remain effective. The resulting k-adjacency matrices are also more sparse than their exponentiated counterparts (see Fig. 2), allowing more efficient representations.

3.3. G3D: Unified Spatial-Temporal Modeling

Most existing work treats skeleton actions as a sequence of disjoint graphs where features are extracted through spatial-only (*e.g.* GCNs) and temporal-only (*e.g.* TCNs) modules. We argue that such factorized formulation is less effective for capturing complex spatial-temporal joint relationships. Clearly, if a strong connection exists between a pair of nodes, then during layer-wise propagation the pair should incorporate a significant portion each other's features to reflect such a connection [50, 33, 34]. However, as signals are propagated across spacetime through a series of local aggregators (GCNs and TCNs alike), they are weakened as redundant information is aggregated from an increasingly larger spatial-temporal receptive field. The problem is more evident if one observes that GCNs do not perform a weighted aggregation to distinguish each neighbor.

Cross-Spacetime Skip Connections. To tackle the above problem, we propose a more reasonable approach to allow cross-spacetime *skip connections*, which are readily modeled with cross-spacetime *edges* in a spatial-temporal graph. Let us first consider a sliding temporal window of size τ over the input graph sequence, which, at each step, obtains a spatial-temporal subgraph $\mathcal{G}_{(\tau)} = (\mathcal{V}_{(\tau)}, \mathcal{E}_{(\tau)})$ where $\mathcal{V}_{(\tau)} = \mathcal{V}_1 \cup ... \cup \mathcal{V}_\tau$ is the union of all node sets across τ frames in the window. The initial edge set $\mathcal{E}_{(\tau)}$ is defined by tiling $\tilde{\mathbf{A}}$ into a block adjacency matrix $\tilde{\mathbf{A}}_{(\tau)}$, where

$$\tilde{\mathbf{A}}_{(\tau)} = \begin{bmatrix} \tilde{\mathbf{A}} & \cdots & \tilde{\mathbf{A}} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}} & \cdots & \tilde{\mathbf{A}} \end{bmatrix} \in \mathbb{R}^{\tau N \times \tau N}. \quad (5)$$

Intuitively, each submatrix $[\tilde{\mathbf{A}}_{(\tau)}]_{i,j} = \tilde{\mathbf{A}}$ means every node in $\mathcal{V}_i$ is connected to itself and its 1-hop spatial neighbors at frame j by *extrapolating* the frame-wise spatial connectivity (which is $[\tilde{\mathbf{A}}_{(\tau)}]_{i,i}$ for all i) to the temporal domain. Thus, each node within $\mathcal{G}_{(\tau)}$ is densely connected to itself and its 1-hop spatial neighbors across all τ frames. We can easily obtain $\mathbf{X}_{(\tau)} \in \mathbb{R}^{T \times \tau N \times C}$ using the same sliding window over $\mathbf{X}$ with zero padding to construct T windows. Using Eq. 1, we thus arrive at a *unified* spatial-temporal graph convolutional operator for the tth temporal window:

$$[\mathbf{X}_{(\tau)}^{(l+1)}]_t = \sigma\left(\tilde{\mathbf{D}}_{(\tau)}^{-\frac{1}{2}} \tilde{\mathbf{A}}_{(\tau)} \tilde{\mathbf{D}}_{(\tau)}^{-\frac{1}{2}} [\mathbf{X}_{(\tau)}^{(l)}]_t \Theta^{(l)}\right). \quad (6)$$

Dilated Windows. Another significant aspect of the above window construction is that the frames need not to be adjacent. A *dilated* window with τ frames and a dilation rate d can be constructed by picking a frame every d frames, and reusing the same spatial-temporal structure $\tilde{\mathbf{A}}_{(\tau)}$. Similarly, we can obtain node features $\mathbf{X}_{(\tau,d)} \in \mathbb{R}^{T \times \tau N \times C}$ ($d = 1$ if omitted) and perform layer-wise update as in Eq. 6. Dilated windows allow larger temporal receptive fields *without* growing the size of $\tilde{\mathbf{A}}_{(\tau)}$, analogous to how dilated convolutions [53] keep constant complexities.

Multi-Scale G3D. We can also integrate the proposed disentangled multi-scale aggregation scheme (Eq. 4) into G3D for multi-scale reasoning *directly* in the spatial-temporal domain. We thus derive the MS-G3D module from Eq. 6 as:

$$[\mathbf{X}_{(\tau)}^{(l+1)}]_t = \sigma \left(\sum_{k=0}^{K} \tilde{\mathbf{D}}_{(\tau,k)}^{-\frac{1}{2}} \tilde{\mathbf{A}}_{(\tau,k)} \tilde{\mathbf{D}}_{(\tau,k)}^{-\frac{1}{2}} [\mathbf{X}_{(\tau)}^{(l)}]_t \Theta_{(k)}^{(l)} \right), \quad (7)$$

where $\tilde{\mathbf{A}}_{(\tau,k)}$ and $\tilde{\mathbf{D}}_{(\tau,k)}$ are defined similarly as $\tilde{\mathbf{A}}_{(k)}$ and $\tilde{\mathbf{D}}_{(k)}$ respectively. Remarkably, our proposed disentangled aggregation scheme complements this unified operator, as G3D's increased node degrees from spatial-temporal connectivity can contribute to the biased weighting problem.

Discussion. We give more in-depth analyses on G3D as follows. (1) It is analogous to classical 3D convolutional blocks [38], with its spatial-temporal receptive field defined by τ, d, and $\tilde{\mathbf{A}}$. (2) Unlike 3D convolutions, G3D's parameter count from $\Theta_{(\cdot)}^{(\cdot)}$ is independent of τ or $|\mathcal{E}_{(\tau)}|$, making it generally less prone to overfitting with large τ. (3) The dense cross-spacetime connections in G3D entail a trade-off on τ, as larger values of τ bring larger temporal receptive fields at the cost of more *generic* features due to larger immediate neighborhoods. Additionally, larger τ implies a quadratically larger $\tilde{\mathbf{A}}_{(\tau)}$ and thus more operations with multi-scale aggregation. On the other hand, larger dilations d bring larger temporal coverage at the cost of temporal *resolution* (lower frame rates). τ and d thus must be balanced carefully. (4) G3D modules are designed to capture complex *regional* spatial-temporal instead of long-range dependencies that are otherwise more economically captured by factorized modules. We thus observe the best performance when G3D modules are augmented with long-range, factorized modules, which we discuss in the next section.

3.4. Model Architecture

Overall Architecture. The final model architecture is illustrated in Fig. 3. On a high level, it contains a stack of r spatial-temporal graph convolutional (STGC) blocks to extract features from skeleton sequences, followed by a global average pooling layer and a softmax classifier. Each STGC block deploys two types of pathways to simultaneously capture complex regional spatial-temporal joint corre-

lations as well as long-range spatial and temporal dependencies: (1) The G3D pathway first constructs spatial-temporal windows, performs disentangled multi-scale graph convolutions on them, and then *collapses* them with a fully connected layer for window feature readout. The extra dotted G3D pathway (Fig. 3(b)) indicates the model can learn from multiple spatial-temporal contexts concurrently with different τ and d; (2) The factorized pathway augments the G3D pathway with long-range, spatial-only, and temporal-only modules: the first layer is a multi-scale graph convolutional layer capable of modeling the entire skeleton graph with the maximum K; it is then followed by two multi-scale temporal convolutions layers to capture extended temporal contexts (discussed below). The outputs from all pathways are aggregated as the STGC block output, which has 96, 192, and 384 feature channels respectively within a typical $r=3$ block architecture. Batch normalization [14] and ReLU is added at the end of each layer except for the last layer. All STGC blocks, except the first, downsample the temporal dimension with stride 2 temporal conv and sliding windows.

Multi-Scale Temporal Modeling. The spatial-temporal windows $\mathcal{G}_{(\tau)}$ used by G3D are a closed structure by themselves, which means G3D must be accompanied by temporal modules for cross-window information exchange. Many existing work [50, 18, 33, 32, 21] performs temporal modeling using temporal convolutions with a fixed kernel size $k_t \times 1$ throughout the architecture. As a natural extension to our multi-scale spatial aggregation, we enhance vanilla temporal convolutional layers with multi-scale learning, as illustrated in Fig. 3(c). To lower the computational costs due to the extra branches, we deploy a bottleneck design [37], fix kernel sizes at 3×1, and use different dilation rates [53] instead of larger kernels for larger receptive fields. We also use residual connections [12] to facilitate training.

Adaptive Graphs. To improve the flexibility of graph convolutional layers which performs homogeneous neighborhood averaging, we add a simple learnable, unconstrained graph residual mask $\mathbf{A}^{\text{res}}$ inspired by [33, 32] to every $\tilde{\mathbf{A}}_{(k)}$ and $\tilde{\mathbf{A}}_{(\tau,k)}$ to strengthen, weaken, add, or remove edges dynamically. For example, Eq. 4 is updated to

$$\mathbf{X}_t^{(l+1)} = \sigma \left(\sum_{k=0}^{K} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} (\tilde{\mathbf{A}}_{(k)} + \mathbf{A}_{(k)}^{\text{res}}) \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \mathbf{X}_t^{(l)} \Theta_{(k)}^{(l)} \right). \quad (8)$$

$\mathbf{A}^{\text{res}}$ is initialized with random values around zero and is different for each k and τ, allowing each multi-scale context (either spatial or spatial-temporal) to select the best suited mask. Note also that since $\mathbf{A}^{\text{res}}$ is optimized for all possible actions, which may have different optimal edge sets for feature propagation, it is expected to give minor edge corrections and may be insufficient when the graph structures have major deficiencies. In particular, $\mathbf{A}^{\text{res}}$ only partially mitigates the biased weighting problem (see Section 4.3).
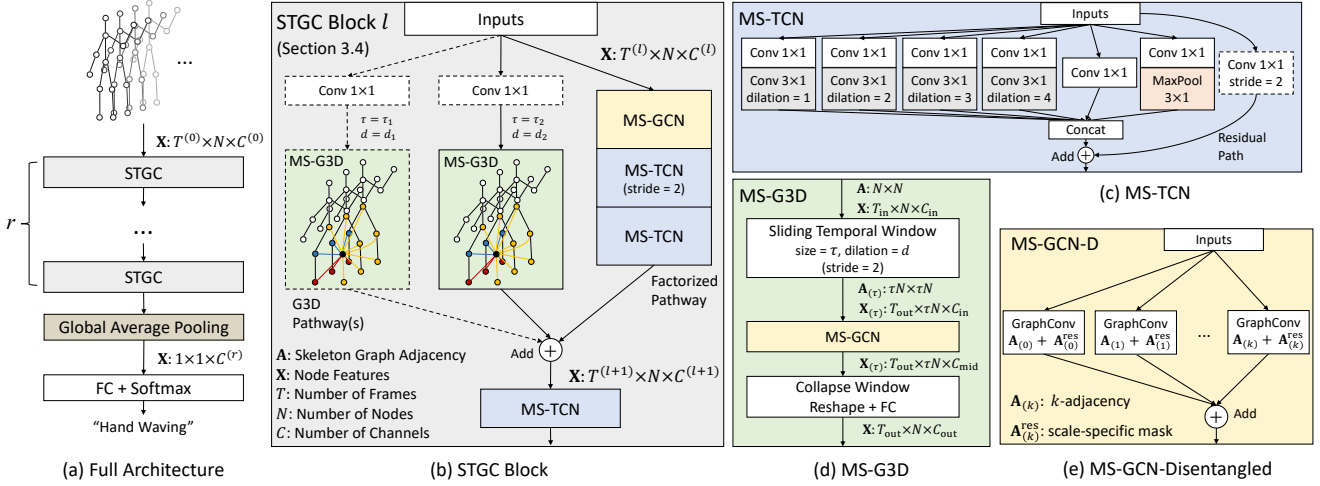
Figure 3: (Match components with colors) **Architecture Overview**. "TCN", "GCN", prefix "MS-", and suffix "-D" denotes temporal and graph convolutional blocks, and multi-scale and disentangled aggregation, respectively (Section 3.2). Each of the r STGC blocks (b) deploys a multi-pathway design to capture long-range and regional spatial-temporal dependencies simultaneously. **Dotted modules**, including extra G3D pathway, 1×1 conv, and strided temporal convolutions, are situational for model performance/complexity trade-off.

Joint-Bone Two-Stream Fusion. Inspired by the two-stream methods in [33, 32, 34] and the intuition that visualizing *bones* along with joints can help humans recognize skeleton actions, we use a two-stream framework where a *separate* model with identical architecture is trained using the bone features initialized as vector differences of adjacent joints directed away from the body center. The softmax scores from the joint/bone models are summed to obtain final prediction scores. Since skeleton graphs are *trees*, we add a zero bone vector at the body center to obtain N bones from N joints and reuse $\mathbf{A}$ for connectivity definition.

4. Experiments

4.1. Datasets

NTU RGB+D 60 and NTU RGB+D 120. NTU RGB+D 60 [31] is a large-scale action recognition dataset containing 56,578 skeleton sequences over 60 action classes captured from 40 distinct subjects and 3 different camera view angles. Each skeleton graph contains $N = 25$ body joints as nodes, with their 3D locations in space as initial features. Each frame of the action contains 1 to 2 subjects. The authors recommend reporting the classification accuracy under two settings: (1) Cross-Subject (X-Sub), where the 40 subjects are split into training and testing groups, yielding 40,091 and 16,487 training and testing examples respectively. (2) Cross-View (X-View), where all 18,932 samples collected from camera 1 are used for testing and the rest 37,646 samples used for training. NTU RGB+D 120 [25] extends NTU RGB+D 60 with an additional 57,367 skeleton sequences over 60 extra action classes, totalling 113,945 samples over 120 classes captured from 106 distinct sub-

jects and 32 different camera setups. The authors now recommend replacing the Cross-View setting with a Cross-Setup (X-Set) setting, where 54,468 samples collected from half of the camera setups are used for training and the rest 59,477 samples for testing. In Cross-Subject, 63,026 samples from a selected group of 53 subjects are used for training, and the rest 50,919 samples for testing.

Kinetics Skeleton 400. The Kinetics Skeleton 400 dataset is adapted from the Kinetics 400 video dataset [15] using the OpenPose [4] pose estimation toolbox. It contains 240,436 training and 19,796 testing skeleton sequences over 400 classes, where each skeleton graph contains 18 body joints, along with their 2D spatial coordinates and the prediction confidence score from OpenPose as the initial joint features [50]. At each time step, the number of skeletons is capped at 2, and skeletons with lower overall confidence scores are discarded. Following the convention from [15, 50], Top-1 and Top-5 accuracies are reported.

4.2. Implementation Details

Unless otherwise stated, all models have $r = 3$ and are trained with SGD with momentum 0.9, batch size 32 (16 per worker), an initial learning rate 0.05 (can linearly scale up with batch size [9]) for 50, 60, and 65 epochs with step LR decay with a factor of 0.1 at epochs {30, 40}, {30, 50}, and {45, 55} for NTU RGB+D 60, 120, and Kinetics Skeleton 400, respectively. Weight decay is set to 0.0005 for final models and is adjusted accordingly during component studies. All skeleton sequences are padded to $T = 300$ frames by replaying the actions. Inputs are preprocessed with normalization and translation following [33, 32]. No data augmentation is used for fair performance comparison.

4.3. Component Studies

We analyze the individual components and their configurations in the final architecture. Unless stated, performance is reported as classification accuracy on the Cross-Subject setting of NTU RGB+D 60 using only the joint data.

Disentangled Multi-Scale Aggregation. We first justify our proposed disentangled multi-scale aggregation scheme by verifying its effectiveness with different number of scales over sparse and dense graphs. In Table 1, we do so using the individual pathways of the STGC blocks (Fig. 3(b)), referred to as "GCN" and "G3D", respectively, with suffixes "-E" and "-D" denoting adjacency powering and disentangled aggregation. Here, the maximum $K = 12$ is the diameter of skeleton graphs from NTU RGB+D 60, and we set $\tau = 5$ for G3D modules. To keep consistent normalization, we set $\widehat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ in Eq. 2 for GCN-E and G3D-E. We first observe that the disentangled formulation can bring as much as 1.4% gain over simple adjacency powering at $K = 4$, underpinning the necessity for neighborhood disentanglement. In this case, the residual mask $\mathbf{A}^{\text{res}}$ partially corrects the weighting imbalance, narrowing the largest gap to 0.4%. However, the same set of experiments on the G3D pathway, where the window graph $\mathcal{G}_{(\tau)}$ is denser than the spatial graph $\mathcal{G}$, shows wider accuracy gaps between G3D-E and G3D-D, indicating a more severe biased weighting problem. In particular, we see 0.8% performance gap at $K = 12$ even if residual masks are added. These results verify the effectiveness of the proposed disentangled aggregation scheme for multi-scale learning; it boosts performance across different number scales not only in the spatial domain, but more so in the spatial-temporal domain where it complements the proposed G3D module. In general, the spatial GCNs benefits more from large K than do the spatial-temporal G3D modules; for final architectures, we empirically set $K \in \{12, 5\}$ for MS-GCN and MS-G3D blocks respectively.

Effectiveness of G3D. To validate the efficacy of G3D modules to capture complex spatial-temporal features, we build up the model incrementally with its individual components, and show its performance in Table 2. We use the joint stream from 2s-AGCN [33] as the baseline for controlled experiments, and for fair comparison, we replaced its regular temporal convolutional layers with MS-TCN layers and obtained an improvement with less parameters. First, we observe that the factorized pathway alone can outperform the baseline due to the powerful disentangled aggregation in MS-GCN. However, if we simply scale up the factorized pathway to larger capacity (deeper and wider), or duplicate the factorized pathway to learn from different feature subspaces and mimic the multi-pathway design in STGC blocks, we observe limited gains. In contrast, when

| Methods | Number of Scales | | | |
	$K=1$	$K=4$	$K=8$	$K=12$
GCN-E	85.1	85.6	86.5	86.6
GCN-D	85.1	87.0	86.9	86.8
GCN-E + Mask	86.1	87.0	87.5	87.7
GCN-D + Mask	86.1	86.9	87.9	87.8
G3D-E	85.1	85.5	85.4	85.5
G3D-D	85.1	86.4	86.5	86.4
G3D-E + Mask	86.6	87.0	86.5	86.2
G3D-D + Mask	86.6	87.4	87.1	87.0

Table 1: Accuracy (%) with multi-scale aggregation on individual pathways of STGC blocks with different K. "Mask" refers to the residual masks $\mathbf{A}^{\text{res}}$. If $K > 1$, GCN/G3D is **Multi-Scale (MS-)**.

Model Configurations	Params	Acc (%)
Baseline (Js-AGCN [33])	3.5M	86.0
Baseline + MS-TCN	1.6M	86.7
MS-GCN (Factorized Pathway) Only	1.4M	87.8
with $2.5\times$ Capacity	3.5M	88.5
with Dual Pathway	2.8M	88.6
MS-GCN (Factorized Pathway)		
with MS-G3D ($\tau = 3, d = 1$)	2.7M	89.0
with MS-G3D ($\tau = 3, d = 2$)	2.7M	89.1
with MS-G3D ($\tau = 3, d = 3$)	2.7M	89.1
with MS-G3D ($\tau = 5, d = 1$)	3.2M	89.2
with MS-G3D ($\tau = 5, d = 2$)	3.2M	89.2
with MS-G3D ($\tau = 7, d = 1$)[†]	3.0M	89.0
with 2 MS-G3D Pathways[†] $\tau = (3,3), d = (1,2)$	2.8M	89.3
with 2 MS-G3D Pathways[†] $\tau = (3,5), d = (1,1)$	3.2M	89.4

Table 2: Model accuracy with various settings. MS-GCN and MS-G3D uses $K \in \{12, 5\}$ respectively. [†]Output channels double at the collapse window layer (Fig. 3(d), C_{mid} to C_{out}) instead of at the graph convolution (C_{in} to C_{mid}) to maintain similar budget.

G3D Graph Connectivity	Params	Acc (%)
(1) Grid-like	2.7M	88.7
(2) Grid-like + dense self-edges	2.7M	88.6
(Eq. 5) Cross-spacetime edges	2.7M	89.1

Table 3: Comparing graph connectivity settings ($\tau = 3, d = 2$).

the G3D pathway is added, we observe consistently better results with similar or less parameters, verifying G3D's ability to pick up complex regional spatial-temporal correlations that are previously overlooked by modeling spatial and temporal dependencies in a factorized fashion.

Exploring G3D Configurations. Table 2 also compares various G3D settings, including different values of τ, d, and the number of G3D pathways in STGC blocks. We

Methods	NTU RGB+D 120	
	X-Sub (%)	X-Set (%)
ST-LSTM [26]	55.7	57.9
GCA-LSTM [27]	61.2	63.3
RotClips + MTCNN [16]	62.2	61.8
Body Pose Evolution Map [28]	64.6	66.9
2s-AGCN [33]	82.9	84.9
MS-G3D Net	**86.9**	**88.4**

Table 4: Classification accuracy comparison against state-of-the-art methods on the NTU RGB+D 120 Skeleton dataset.

Methods	NTU RGB+D 60	
	X-Sub (%)	X-View (%)
IndRNN [23]	81.8	88.0
HCN [20]	86.5	91.1
ST-GR [18]	86.9	92.3
AS-GCN [21]	86.8	94.2
2s-AGCN [33]	88.5	95.1
AGC-LSTM [34]	89.2	95.0
DGNN [32]	89.9	96.1
GR-GCN [8]	87.5	94.3
MS-G3D Net (Joint Only)	89.4	95.0
MS-G3D Net (Bone Only)	90.1	95.3
MS-G3D Net	**91.5**	**96.2**

Table 5: Classification accuracy comparison against state-of-the-art methods on the NTU RGB+D 60 Skeleton dataset.

Methods	Kinetics Skeleton 400	
	Top-1 (%)	Top-5 (%)
ST-GCN [50]	30.7	52.8
AS-GCN [21]	34.8	56.5
ST-GR [18]	33.6	56.1
2s-AGCN [33]	36.1	58.7
DGNN [32]	36.9	59.6
MS-G3D Net	**38.0**	**60.9**

Table 6: Classification accuracy comparison against state-of-the-art methods on the Kinetics Skeleton 400 dataset.

first observe that all configurations consistently outperform the baseline, confirming the stability of MS-G3D as a robust feature extractor. We also see that $\tau = 5$ give slightly better results, but the gain diminishes at $\tau = 7$ as the aggregated features become too generic due to the oversized local spatial-temporal neighborhood, thus counteracting the benefits of larger temporal coverage. The dilation rate d has varying effects: (1) when $\tau = 3$, $d = 1$ underperforms $d \in \{2, 3\}$, justifying the need for larger temporal contexts; (2) larger d has marginal benefits, as its larger temporal coverage come at a cost of temporal resolution (thus *coarsened* skeleton motions). We thus observe better results when *two* G3D pathways with $d = (1, 2)$ are combined, and as ex-

pected, we obtain the best results when the temporal resolution is unaltered by setting $\tau = (3, 5)$.

Cross-spacetime Connectivity. To demonstrate the need for cross-spacetime edges in $\mathcal{G}_{(\tau)}$ defined in Eq. 5 instead of simple, *grid-like* temporal self-edges (on which G3D also applies), we contrast different connectivity schemes in Table 3 while fixing other parts of the architecture. The first two settings refer to modifying the block adjacency matrix $\tilde{\mathbf{A}}_{(\tau)}$ such that: (1) the blocks $\tilde{\mathbf{A}}$ on the main diagonal are kept, the blocks on superdiagonal/subdiagonal is set to $\mathbf{I}$, and the rest set to $\mathbf{0}$; and (2) all blocks but the main diagonal of $\tilde{\mathbf{A}}$ are set to $\mathbf{I}$. Intuitively, the first produces "3D grid" graphs and the second includes extra dense self-edges over τ frames. Clearly, while all settings allow unified spatial-temporal graph convolutions, cross-spacetime edges as skip connections are essential for efficient information flow.

Joint-Bone Two-Stream Fusion. We verify our method under the joint-bone fusion framework on the NTU RGB+D 60 dataset in Table 5. Similar to [33], we obtain best performance when joint and bone features are fused, indicating the generalizablity of our method to other input modalities.

4.4. Comparison against the State-of-the-Art

We compare our full model (Fig. 3(a)) to the state-of-the-art in Tables 4, 5, and 6. Table 4 compares non-graph [26, 27, 16, 28] and graph-based methods [33]. Table 5 compares non-graph methods [23, 20], graph-based methods with spatial edges [18, 21, 33, 34, 32] and with spatial-temporal edges [8]. Table 6 compares single-stream [50, 21] and multi-stream [18, 33, 32] methods. On all three large-scale datasets, our method outperforms all existing methods under all evaluation settings. Notably, our method is the first to apply a *multi-pathway* design to learn both long-range spatial and temporal dependencies and complex regional spatial-temporal correlations from skeleton sequences, and the results verify the effectiveness of our approach.

5. Conclusion

In this work, we present two methods for improving skeleton-based action recognition: a disentangled multi-scale aggregation scheme for graph convolutions that removes redundant dependencies between different neighborhoods, and G3D, a unified spatial-temporal graph convolutional operator that directly models spatial-temporal dependencies from skeleton graph sequences. By coupling these methods, we derive MS-G3D, a powerful feature extractor that captures multi-scale spatial-temporal features previously overlooked by factorized modeling. With experiments on three large-scale datasets, we show that our model outperforms existing methods by a sizable margin.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 21–29, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 2

[2] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1993–2001, 2016. 2

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 6

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 2

[6] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 1, 3

[7] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2083–2092, 2019. 2

[8] Xiang Gao, Wei Hu, Jiaxiang Tang, Jiaying Liu, and Zongming Guo. Optimized skeleton-based action recognition via sparsified graph regression. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 601–610, New York, NY, USA, 2019. ACM. 3, 8

[9] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017. 2

[11] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[13] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data, 2015. 2

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[15] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 1, 2, 6

[16] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. Learning clip representations for skeleton-based 3d action recognition. *IEEE Transactions on Image Processing*, 27(6):2842–2855, 2018. 8

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 2, 4

[18] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. Spatiotemporal graph routing for skeleton-based action recognition. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019. 2, 3, 5, 8

[19] Chaolong Li, Zhen Cui, Wenming Zheng, Chunyan Xu, and Jian Yang. Spatio-temporal graph convolution for skeleton based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 3

[20] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jul 2018. 1, 8

[21] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019. 1, 2, 3, 5, 8

[22] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[23] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 8

[24] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S Zemel. Lanczosnet: Multi-scale deep graph convolutional networks. *arXiv preprint arXiv:1901.01484*, 2019. 2, 3

[25] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 11, 2019. 1, 2, 6

[26] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016. 8

[27] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017. 8

[28] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1159–1168, 2018. 8

[29] Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. *arXiv*, 1906.02174, 2019. 2

[30] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatiotemporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017. 2

[31] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016. 1, 2, 3, 6

[32] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 1, 2, 3, 5, 6, 7, 8

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[34] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 1, 2, 3, 4, 6, 8

[35] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 1

[36] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 3

[37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016. 5

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 5

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[41] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018. 2

[42] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014. 1, 3

[43] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297. IEEE, 2012. 1, 3

[44] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018. 2

[45] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 2

[46] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019. 2

[47] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019. 2

[48] Chunyu Xie, Ce Li, Baochang Zhang, Chen Chen, Jungong Han, and Jianzhuang Liu. Memory attention networks for skeleton-based action recognition. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jul 2018. 1, 3

[49] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019. 2

[50] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 3, 4, 5, 6, 8

[51] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pages 4800–4810, 2018. 2

[52] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017. 2

[53] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions, 2015. 5

[54] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017. 1, 3