Decoupling GCN with DropGraph Module for Skeleton-Based Action Recognition

Ke Cheng[1,2], Yifan Zhang[1,2] ✉, Congqi Cao[4], Lei Shi[1,2], Jian Cheng[1,2,3], and Hanqing Lu[1,2]

[1] NLPR & AIRIA, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] CAS Center for Excellence in Brain Science and Intelligence Technology
[4] School of Computer Science, Northwestern Polytechnical University
chengke2017@ia.ac.cn, {yfzhang, lei.shi, jcheng, luhq}@nlpr.ia.ac.cn, congqi.cao@nwpu.edu.cn

Abstract. In skeleton-based action recognition, graph convolutional networks (GCNs) have achieved remarkable success. Nevertheless, how to efficiently model the spatial-temporal skeleton graph without introducing extra computation burden is a challenging problem for industrial deployment. In this paper, we rethink the spatial aggregation in existing GCN-based skeleton action recognition methods and discover that they are limited by coupling aggregation mechanism. Inspired by the decoupling aggregation mechanism in CNNs, we propose decoupling GCN to boost the graph modeling ability with no extra computation, no extra latency, no extra GPU memory cost, and less than 10% extra parameters. Another prevalent problem of GCNs is over-fitting. Although dropout is a widely used regularization technique, it is not effective for GCNs, due to the fact that activation units are correlated between neighbor nodes. We propose DropGraph to discard features in correlated nodes, which is particularly effective on GCNs. Moreover, we introduce an attention-guided drop mechanism to enhance the regularization effect. All our contributions introduce zero extra computation burden at deployment. We conduct experiments on three datasets (NTU-RGBD, NTU-RGBD-120, and Northwestern-UCLA) and exceed the state-of-the-art performance with less computation cost.

Keywords: skeleton-based action recognition, decoupling GCN, DropGraph

1 Introduction

Human action recognition, which plays an essential role in video understanding and human-computer interaction, attracts more attention in recent years [43,39,46]. Compared with action recognition with RGB video, skeleton-based action recognition is robust to circumstance changes and illumination variations [3,40,36,48,50,25,46,17,33,34,29].

Traditional methods mainly focus on designing hand-crafted features [39,4]. However, the performance of these handcrafted-features-based methods is barely satisfactory. Deep learning methods usually rearrange a skeleton sequence as a pseudo-image [21,14,9] or a series of joint coordinates [32,50,25], then use CNNs or RNNs to predict action labels. Recently, graph convolutional networks (GCNs), which generalize CNNs from image to graph, have been successfully adopted to model skeleton data [46]. The key component of GCN is spatial aggregation, which aggregates features of different body joints. To increase the flexibility of the skeleton graph construction, researchers propose various modules to enhance the spatial aggregation ability for GCNs [44,34,13,35,18].

In this paper, we rethink the spatial aggregation of GCNs, which is derived from CNNs. We discover that existing GCN-based skeleton action recognition methods neglect an important mechanism in CNNs: *decoupling aggregation*. Concretely, every channel has an independent spatial aggregation kernel in CNNs, capturing different spatial information in different frequencies, orientations and colors, which is crucial for the success of CNNs. However, all the channels in a graph convolution share one spatial aggregation kernel: the adjacent matrix. Although some researchers partition one adjacent matrix into multiple adjacent matrices and ensemble multiple graph convolution results of these adjacent matrices [46,34,33,17,13,35,44], the number of adjacent matrices is typically less than 3, which limits the expressiveness of spatial aggregation. Increasing the number of adjacent matrices will cause multiplying growth of computation cost and reduce efficiency.

Inspired by the decoupling aggregation in CNNs, we propose DeCoupling Graph Convolutional Networks (DC-GCN) to address the above dilemma with no extra FLOPs, latency, and GPU memory. DC-GCN split channels into g decoupling groups, and each group has a trainable adjacent matrix, which largely increases the expressiveness of spatial aggregation. Note that the FLOPs of decoupling graph convolution is exactly the same with conventional graph convolution. More importantly, DC-GCN is hardware-friendly and increases no extra time and GPU memory, the two most determining factors in industrial deployment. Besides, DC-GCN only cost $5\% \sim 10\%$ extra parameters.

Another prevalent problem in graph convolution is over-fitting. Although dropout [37] is widely used in GCNs, we discover that the performance does not increase obviously with dropout layer. Because graph convolution is actually a special form of Laplacian smoothing [19], activation units are correlated between neighbor nodes. Even one node in a graph is dropped, information about this node can still be obtained from its neighbor nodes, which causes over-fitting. To relieve the over-fitting problem, we propose DropGraph, a particularly effective regularization method for graph convolutional networks. The key idea is: when we drop one node, we drop its neighbor node together. In addition, we propose an attention-guided drop mechanism to enhance the regularization effect.

The main contributions of this work are summarized as follows: 1) We propose DC-GCN, which efficiently enhances the expressiveness of graph convolution with zero extra computation cost. 2) We propose ADG to effectively relieve

the crucial over-fitting problem in GCNs. 3) Our approach exceeds the state-of-the-art method with less computation cost. Code will be available at `https://github.com/kchengiva/DecoupleGCN-DropGraph`.

2 Background

Human skeleton graph convolution The skeleton data represents a human action as multiple skeleton frames. Every skeleton frame is represented as a graph $\mathcal{G}(V, E)$, where V is the set of n body joints and E is a set of m bones. For example, 3D joints skeleton positions across T frames can be represented as $\mathcal{X} \in \mathbb{R}^{n\times 3\times T}$, and the 3D joints skeleton frames in the t-th frame is denoted as $\mathbf{X}_t = \mathcal{X}_{:,:,t} \in \mathbb{R}^{n\times 3}$. GCN-based action recognition models [46,44,34,33,13,35,18] are composed of several spatial-temporal GCN blocks, where spatial graph convolution is the key component.

Let $\mathbf{X} \in \mathbb{R}^{n\times C}$ be the input features in one frame, and $\mathbf{X}' \in \mathbb{R}^{n\times C'}$ be the output features of these joints, where C and C' are the input and output feature dimension respectively. The spatial graph convolution is

$$\mathbf{X}' = \sum_{p\in\mathcal{P}} \mathbf{X}'^{(p)} = \sum_{p\in\mathcal{P}} \widetilde{\mathbf{A}^{(p)}}\mathbf{X}\mathbf{W}^{(p)}, \tag{1}$$

where $\mathcal{P}$ ={root,centripetal,centrifugal} denotes the partition subsets [46]. $\widetilde{\mathbf{A}^{(p)}}$ is initialized as $\mathbf{D}^{(p)^{-\frac{1}{2}}}\mathbf{A}^{(p)}\mathbf{D}^{(p)^{-\frac{1}{2}}}$, where $\mathbf{D}^{(p)}_{ii} = \sum_j(\mathbf{A}^{(p)}_{ij}) + \varepsilon$. Here ε is set to 0.001 to avoid empty rows. Recent works let both $\widetilde{\mathbf{A}^{(p)}} \in \mathbb{R}^{n\times n}$ and $\mathbf{W}^{(p)} \in \mathbb{R}^{C\times C'}$ trainable [34,33].

Regularization method

Over-fitting is a crucial problem in deep neural networks, including GCNs. Dropout [37] is a common regularization method. Although dropout is very effective at regularizing fully-connected layers, it is not powerful when used in convolutional layers. [2] proposed Cutout for regularizing CNNs, which randomly removes contiguous region in the input images. [5] proposed DropBlock, which applying Cutout at every feature map in CNNs. The reason why Cutout and DropBlock are efficient is that features are spatially correlated in CNNs. GCNs have the similar problems with CNNs, where common dropout is not effective. Inspired from DropBlock [5] in CNNs, we proposed DropGraph to effectively regularize GCNs.

3 Approach

In this section, we analyze the limitation of the human skeleton graph convolutional networks and propose DeCoupling Graph Convolutional Network (DC-GCN). In addition, we propose an attention-guided DropGraph (ADG) to relieve the prevalent overfitting problem in GCNs.

3.1 Decoupling graph convolutional network

For clarity, we first discuss the case of graph convolution with a single partition set, then naturally extend to the multiple partition case.

Motivation Graph convolution contains two matrix multiplication processes: $\widetilde{\mathbf{A}}\mathbf{X}$ and $\mathbf{X}\mathbf{W}$. $\widetilde{\mathbf{A}}\mathbf{X}$ computes the aggregation information between different skeletons, so we call it *spatial aggregation*. $\mathbf{X}\mathbf{W}$ compute the correlate information between different channels, so we call it *channel correlation*.

As shown in Fig.1 (a), the spatial aggregation ($\widetilde{\mathbf{A}}\mathbf{X}$) can be decomposed into computing the aggregation on every channel respectively. Note that all the channels of feature $\mathbf{X}$ share one adjacency matrix $\mathbf{A}$ (drawn in the same color), which means all the channels share the same aggregation kernel. We call it *coupling aggregation*. All existing GCN-based skeleton action recognition methods adopt the *coupling aggregation*, such as ST-GCN [46], Nonlocal adaptive GCN [34], AS-GCN [18], Directed-GNN [33]. We collectively call them *coupling graph convolution*.

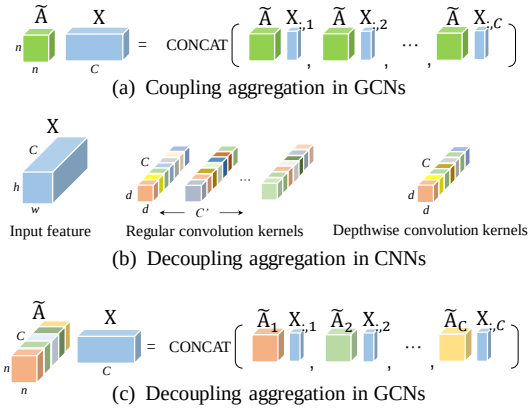


Fig. 1. Conventional GCNs (a) employ coupling aggregation, while CNNs (b) employ decoupling aggregation. We introduce the decoupling aggregation mechanism into GCNs and propose decoupling GCN (c).

However, CNNs, the source of inspiration for GCNs, do not adopt the *coupling aggregation*. As shown in Fig.1 (b), different channels have independent spatial aggregation kernels, shown in different color. We call this mechanism *decoupling aggregation*. Decoupling aggregation mechanism can largely increase the spatial aggregation ability, which is essential for the success of CNNs.

DeCoupling GCN Conventional graph convolution limited by coupling aggregation can be analogous to a "degenerate depthwise convolution" whose convolution kernels are shared between channels. The expressiveness of the "degenerate

depthwise convolution" is notably weaker than a standard depthwise convolution. Therefore, we deduce that existing GCN-based skeleton action recognition models [46,34,18,33,45,35] lack the *decoupling aggregation* mechanism.

In this paper, we propose *decoupling graph convolution* for skeleton action recognition, where different channel has independent trainable adjacent matrix, shown in Fig.1 (c). Decoupling graph convolution largely increases the variety of adjacent matrix. Similar to the redundancy of CNN kernels [30], decoupling graph convolution may introduce redundant adjacent matrix. Hence we split channels into g groups. Channels in a group share one trainable adjacent matrix. When $g = C$, every channel has its own spatial aggregation kernel which causes large number of redundant parameters; when $g = 1$, decoupling graph convolution degenerates into coupling graph convolution. Interestingly, experiments show that $8 \sim 16$ groups are enough. In this case, we only increase $5\% \sim 10\%$ extra parameters. The equation of decoupling graph convolution is shown as below:

$$\mathbf{X}' = \widetilde{\mathbf{A}^d}_{:,:,1}\mathbf{X}^{\mathbf{w}}_{:,\lfloor\frac{\mathbf{C}}{\mathbf{g}}\rfloor} \| \widetilde{\mathbf{A}^d}_{:,:,2}\mathbf{X}^{\mathbf{w}}_{:,\lfloor\frac{\mathbf{C}}{\mathbf{g}}\rfloor:\lfloor\frac{\mathbf{2C}}{\mathbf{g}}\rfloor} \| \cdots \| \widetilde{\mathbf{A}^d}_{:,:,g}\mathbf{X}^{\mathbf{w}}_{:,\lfloor\frac{\mathbf{(g-1)C}}{\mathbf{g}}\rfloor:} \tag{2}$$

where $\mathbf{X}^{\mathbf{w}} = \mathbf{X}\mathbf{W}$, $\widetilde{\mathbf{A}^d} \in \mathbb{R}^{n\times n\times g}$ is the decoupling adjacent matrices. Indexes of $\widetilde{\mathbf{A}^d}$ and $\mathbf{X}^{\mathbf{w}}$ are in Python notation, and $\|$ represents channel-wise concatenation.

By replacing *coupling graph convolution* with *decoupling graph convolution*, we construct DeCoupling GCN (DC-GCN). Although the number of parameters is slightly increased, the floating-number operations (FLOPs) of DC-GCN is exactly the same with conventional GCN ($n^2C + nC^2$). More importantly, DC-GCN costs no extra time and GPU memory, the two determining factors for deployment. Compared with other variants of ST-GCNs [34,33,44,13], DC-GCN achieves higher performance without incurring any extra computations.

Discussion DC-GCN can be naturally extended to multiple partition cases by introducing decoupling graph convolution into every partition. Note that our DC-GCN is different from the multi-partition strategy [46], which ensembles multiple graph convolutions with different adjacent matrices. The FLOPs of the multi-partition strategy is proportional to the number of adjacency matrices, while DC-GCN introduces various adjacency matrices with no extra computation. Besides, all our experiments use 3-partition ST-GCN as baseline, which shows the complementarity between multi-partition strategy and DC-GCN.

DC-GCN is different from SemGCN [49] in many aspects: 1) SemGCN focus on pose regression, while we focus on action recognition. 2) The receptive field of SemGCN is localized, and a heavy non-local module is inserted for non-local modeling. Our DC-GCN has non-local receptive fields and increases no extra FLOPs. 3) The parameter cost of SemGCN is nearly double of baseline. Our DC-GCN only increases $5\% \sim 10\%$ extra parameters.

3.2 Attention-guided DropGraph

Motivation Although dropout is a widely used regularization method, the performance of GCNs does not increase obviously with dropout layer. A possible reason is that graph features are correlated between nearby neighbors. As shown in [19], graph convolution is a special form of Laplacian smoothing, which mixes the features of a node and its neighbors. Even one node is dropped, information about this node can still be obtained from its neighbor node, leading to overfitting. We propose DropGraph to effectively regularize GCNs, and design an attention-guided drop mechanism to further enhance the regularization effect.

DropGraph The main idea of DropGraph is: when we drop one node, we drop its neighbor node set together. DropGraph has two main parameters: γ and K. γ controls the sample probability, and K controls the size of the neighbor set to be dropped. On an input feature map, we first sample root nodes v_{root} with the Bernoulli distribution with probability γ, then drop the activation of v_{root} and the nodes that are at maximum K steps away from v_{root}. DropGraph can be implemented as Algorithm 1.

Algorithm 1 DropGraph

Input: a GCN feature $\mathbf{X} \in \mathbb{R}^{n \times C}$, adjacent matrix $\mathbf{A}$, γ, K, *mode*

if $mode == Inference$ **then**
  return $\mathbf{X}$
else
  Randomly sample $\mathbf{V}_{root} \in \mathbb{R}^n$, every element in $\mathbf{V}_{root}$ is in Bernoulli distribution with probability γ.
  Compute the drop mask $\mathbf{M} \in \mathbb{R}^n$ to mask the nodes that are at maximum K steps away from $\mathbf{V}_{root}$:
  $\mathbf{M} = 1 - \mathbf{Bool}((\mathbf{A}+\mathbf{I})^K \mathbf{V}_{root}^\top)$, where **Bool** is function setting non-zero element to 1.
  Apply the mask: $\mathbf{X} = \mathbf{X} \times \mathbf{M}$
  Normalize the feature:
  $\mathbf{X} = \mathbf{X} \times count(\mathbf{M}) / count\_ones(\mathbf{M})$
end if

Let *keep_prob* denote the probability of an activation unit to be kept. For conventional dropout, $keep\_prob = 1-\gamma$. But for DropGraph, every zero entry on v_{root} is expanded to its $1^{st}, 2^{ed}, \cdots, K^{th}$-order neighborhood. Thus, *keep_prob* depends on both γ and K. In a graph with n nodes and e edges, we define the average degree of each node as $d_{ave} = 2e/n$. The expectation number of nodes in the i^{th}-order neighborhood of a random sampled node can be estimated as:

$$B_i = d_{ave} \times (d_{ave} - 1)^{i-1} \tag{3}$$

The average expanded drop size is estimated as:

$$drop\_size = 1 + \sum_{i=1}^{K} B_i \tag{4}$$

If we want to keep activation units with the probability of $keep\_prob$, we set:

$$\gamma = \frac{1 - keep\_prob}{drop\_size} \tag{5}$$

Note that there might be some overlaps between drop areas, so this equation is only an approximation. In our experiments, we first estimate the $keep\_prob$ to use (between 0.75-0.95), and then compute γ as Eq.5.

Attention-guided drop mechanism To enhance the regularization effect, we let the attention area have higher probability to sample v_{root}. Let v be a node, γ_v denote the probability of sampling the node v as v_{root}. We modify Eq.5 as:

$$\gamma_v = \widetilde{\alpha_v} \frac{1 - keep\_prob}{drop\_size} = \alpha_v \frac{count(\alpha)}{\sum \alpha} \frac{1 - keep\_prob}{drop\_size} \tag{6}$$

where α is the attention map, $\widetilde{\alpha}$ is the normalized attention map, $count(\alpha)$ is the number of elements in α. To assess the distribution of attention area, a common implicit assumption is that the absolute value of an activation is an indication about the importance of one unit [47]. We follow this assumption and generate α by averaging the absolute value across the channel dimension.

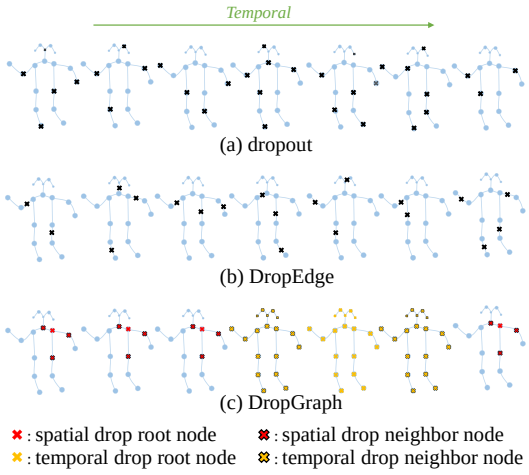


Fig. 2. Spatial-temporal DropGraph.

Spatial-temporal ADG In skeleton action recognition, the input of attention-guided DropGraph (ADG) is a spatiotemporal feature $\mathcal{X} \in \mathbb{R}^{n \times C \times T}$. As shown in Fig.2, we apply ADG to spatial graph and temporal graph respectively.

The spatial aspect of the graph is the human physical structure with the number of nodes n. We generate spatial attention on every skeleton $\alpha_S \in \mathbb{R}^n$ by compressing the absolute value of $\mathcal{X}$ using average pooling on channel dimension and temporal dimension. After sampling v_{root}, we expand the drop area to its spatial neighbors. Then we broadcast the drop area to all temporal frames.

The temporal aspect of the graph is constructed by connecting consecutive frames on temporal dimension, with the number of nodes T. We generate temporal attention on every frame $\alpha_T \in \mathbb{R}^T$ by compressing the absolute value of $\mathcal{X}$ using average pooling on channel dimension and skeleton dimension. After sampling v_{root}, we expand the drop area to its temporal neighbors. Then we broadcast the drop area to all body joints.

We cascade spatial ADG and temporal ADG to construct spatiotemporal ADG. We apply ADG on both GCN branch and skip connection branch. We adopt linear scheme [5] to decreasing *keep_prob* over time from 1 to target value.

Comparison with other regularization methods. We compare DropGraph with other two regularization methods for GCNs: (a) dropout [37], which randomly drops the nodes with a certain probability; (b) DropEdge [31], which randomly drop the edges in a graph with a certain probability. As shown in Fig.2, the drop area of both dropout and DropEdge are isolated, which can not effectively remove related information of the dropped node. For dropout, even if one node is dropped, information about this node can still be obtained from its neighbor node. For DropEdge, even if one edge is dropped, related information can still reach this node through other edges. DropGraph addresses their drawbacks and achieve notably better performance (details in Sec.4.2).

4 Experiments

4.1 Datasets and Model Configuration

NTU-RGBD. NTU-RGBD is the most widely used 3D joint coordinates dataset. It contains 56,880 action samples in 60 action classes. These samples are performed by 40 distinct subjects. The 3D skeleton data is captured by Kinect V2. Each action is captured by 3 cameras from different horizontal angles: $-45°$, $0°$, $45°$. The original paper [32] recommends two protocols. 1) Cross-Subject (X-sub): training data comes from 20 subjects, and the remaining 20 subjects are used for validation. 2) Cross-View (X-view): training data comes from the camera $0°$ and $45°$, and validation data comes from camera $-45°$.

NTU-RGBD-120. NTU-RGBD-120 is the extended version of the NTU-RGBD dataset. It contains 114,480 action samples in 120 action classes, performed by 106 distinct subjects. This dataset contains 32 setups, and every different setup has a specific location and background. The original paper [22]

recommends two evaluation protocols. 1). Cross-Subject (X-sub): training data comes from 53 subjects, and the remaining 53 subjects are used for validation. 2). Cross-Setup (X-setup): picking all the samples with even setup IDs for training, and the remaining samples with odd setup IDs for validation.

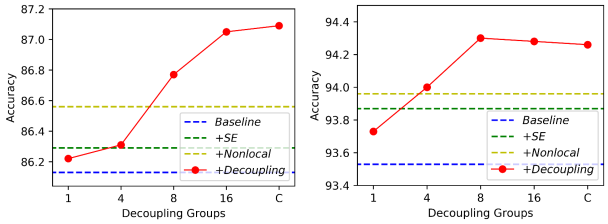
Northwestern-UCLA. Northwestern-UCLA (NW-UCLA) dataset [42] contains 1494 video clips covering 10 categories, which is captured by three Kinect cameras. Each action is performed by 10 different subjects. We adopt the same protocol as [42]: training data comes from the first two cameras, and samples from the other camera are used for validation.

Model Setting. We construct the backbone as ST-GCN [46]. The batch size is 64. We use SGD to train the model for 100 epochs. We use momentum of 0.9 and weight decay of 1e-4. The learning rate is set as 0.1 and is divided by 10 at epoch 60 and 80. For NTU-RGBD and NTU-RGBD-120, we use the same data preprocess as [34]. For NW-UCLA, we use the same data preprocess as [35].

4.2 Ablation Study

Decoupling graph convolution In this subsection, we demonstrate the effectiveness and efficiency of DC-GCN.

(1) Efficacy of DC-GCN. We perform ablation study on different decoupling groups, shown in Fig.3. Our baseline is ST-GCN [46]. We also compare the performance with non-local adaptive graph module (CVPR 2019) [34] and SE module [8]. From Fig.3, we can draw the following conclusions:



(a) X-sub (b) X-view

Fig. 3. Decoupling GCN on NTU-RGBD dataset.

- DC-GCN outperforms the baseline at 1.0% on NTU X-sub task and 0.8% on NTU X-view task. Compared with non-local adaptive graph and SE module, DC-GCN achieves higher performance at no extra computation.
- Compared to coupling graph convolution network (group=1), decoupling graph convolution network achieves higher performance. We do not need to decouple the adjacent matrix of every channel. 8 groups are enough for NTU-RGBD X-view task and 16 groups are enough for NTU-RGBD X-sub task, which is used as our default setting in the following discussion.

– Non-local adaptive GCN [34] uses a non-local module to predict the data-dependent graph for each sample. Compared to it, our DC-GCN employs several static graphs, but get even higher performance with less FLOPs.

Model	GFLOPs	Network Time(ms/batch)	Memory(G)
Baseline	16.2	12	5.014
+SE	16.2	16	5.494
+Nonlocal	17.9	23	5.603
+Coupling $g = 1$	16.2	12	5.014
+Decoupling $g = 4, 8, 16, C$	16.2	12	5.014

Table 1. FLOPs, speed and GPU memory cost. The FLOPs is for one sample on NTU-RGBD dataset. The speed and memory is measured on 1 NVIDIA Tesla K80 GPU with batch size = 64 in PyTorch evaluation mode. The time is network time, without the data loading time.

(2) FLOPs, speed and GPU memory. DC-GCN is not only theoretically efficient but also has high throughput and efficient GPU memory cost, as shown in Table 1. We can conclude that:

– Decoupling GCN ($g = 4, 8, 16, C$) introduces no extra FLOPs/latency/GPU memory compared to coupling GCN ($g = 1$). In addition, the latency and GPU memory cost of DC-GCN are almost the same as ST-GCN baseline.
– DC-GCN is more efficient than non-local adaptive GCN. Non-local adaptive GCN increases 10% extra FLOPs and 589M extra GPU memory, and cost 92 % extra time compared to DC-GCN.
– Although SE module is efficient at FLOPs, it costs 33% extra time and 480M extra GPU memory compared with baseline. This is, the theoretical efficiency is not equivalent to fast speed and efficient GPU memory cost. Compared with SE module, our DC-GCN is a hardware-friendly approach.

(3) Parameter cost. Decoupling GCN introduces extra parameters to baseline. Because our group decoupling mechanism, DC-GCN only introduces $5\% \sim 10\%$ extra parameters when $g = 8 \sim 16$. If we increase the number of channels of baseline to the same parameter cost, we do not get notable improvement.

(4) Visualization of the learned adjacent matrices. We visualize the learned adjacent matrices of coupling GCN (group=1) and decoupling GCN (group=8), shown in Fig.4. Compared with coupling GCN where every channel shares one adjacent matrix, decoupling GCN (group=8) largely increases the variety of spatial aggregation. In this way, decoupling GCN can model diverse relations among joints.

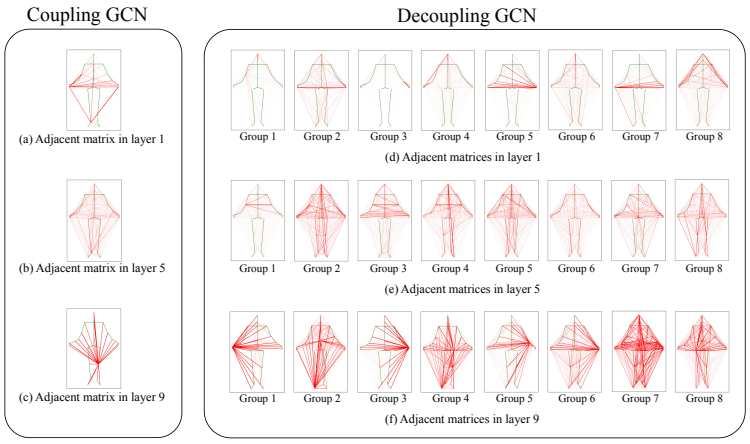


Fig. 4. Visualization of the learned adjacent matrices. The green lines show the body physical connections. The thickness of red lines shows the connection strength of the learned adjacent matrices.

In shallow layers (e.g., layer 1), the skeleton connections in decoupling GCN tend to be local, as shown in Fig.4 (d). For example, some adjacent matrices have strong connections between head and hand (e.g., Group 4 and Group 8 in Fig.4 (d)), which are helpful for recognizing "wipe face" and "brush teeth"; some adjacent matrices have strong connections between head and neck (e.g., Group 1 in Fig.4 (d)), which are helpful for recognizing "nod head" and "shake head"; some adjacent matrices have strong connections between hand and wrist (e.g., Group 3 in Fig.4 (d)), which are helpful for recognizing "write" and "count money" ; some adjacent matrices have strong connections between two hands (e.g., Group 2, Group 5 and Group 7 in Fig.4 (d)), which are helpful for recognizing "clap" and "rub two hands". This characteristic makes decoupling GCN work well in action recognition tasks.

In deep layers (e.g., layer 9), the skeleton connections in decoupling GCN tend to be global, as shown in Fig.4 (f). These adjacent matrices tend to gather the global feature to one joint. In this way, the deep layers can integrate global information (the whole human body) with local information (each single joint), which helps predict the final classification score.

Attention-guided DropGraph In this subsection, we demonstrate the effectiveness of attention-guided DropGraph (ADG).

(1) Comparison with other regularization methods. We compare with three regularization methods: dropout [37], label smoothing [38] and DropEdge [31]. For our proposed DropGraph, we set $K = 1$ for spatial DropGraph and $K = 20$ for temporal DropGraph respectively. The detail ablation study on K is provided in the supplement material. Note that when $K = 0$, DropGraph degenerates to dropout [37]. As shown in Table 2, dropout is not powerful in

GCN. DropGraph notably exceeds the other regularization methods. With the attention-guide drop mechanism, the regularization effect is further enhanced.

Model	Regularization method	X-sub	Δ	X-view	Δ
DC-GCN	-	87.1	0	94.3	0
	dropout [37]	87.2	+0.1	94.4	+0.1
	label smoothing [38]	87.1	+0.0	94.4	+0.1
	DropEdge [31]	87.6	+0.5	94.7	+0.4
	DropGraph (ours)	88.0	+0.9	95.0	+0.7
	Attention-guided DropGraph (ours)	**88.2**	**+1.1**	**95.2**	**+0.9**

Table 2. Compare with other regularization methods. The top-1 accuracy (%) is evaluated on NTU-RGBD. Δ shows the improvement of accuracy.

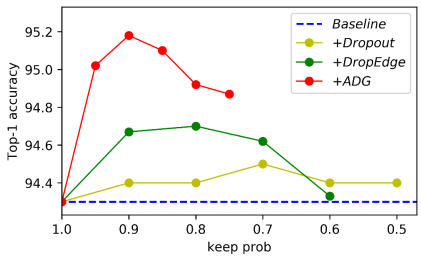


Fig. 5. Compare dropout, DropEdge and our ADG at different *keep_prob*.

(2) The setting of *keep_prob*. We discuss the setting of *keep_prob* on dropout, DropEdge and our proposed ADG. As shown in Fig.5, ADG provides efficient regularization when $keep\_prob = 0.85 \sim 0.9$. ADG has a notable improvement compared to the best result of dropout and DropEdge.

Ablation studies on NW-UCLA and NTU-RGBD-120 Besides the above ablation study on NTU-RGBD dataset, we also perform ablation studies on NW-UCLA and NTU-RGBD-120 datasets, shown in Table 3.

Backbone	+DC	+ADG	NW-UCLA(%)	NTU120 X-sub(%)	NTU120 X-setup(%)
ST-GCN			89.8	79.7	81.3
ST-GCN	✓		91.6	81.3	82.7
ST-GCN	✓	✓	93.8	82.4	84.3

Table 3. Ablation study on NW-UCLA and NTU-RGBD-120. DC refers to decoupling.

4.3 Comparisons to the State-of-the-Art

Multi-stream strategy is commonly employed in previous state-of-the-art approaches [46,34,35,43,33]. We adopt the same multi-stream ensemble strategy with [33], which ensembles 4 streams: joint, bone, motion, and bone motion. The joint stream uses the original joint position as input; the bone stream uses the difference between adjacent joints as input; the motion stream uses the difference between adjacent frames as input.

We conduct extensive experiments on three datasets: NTU-RGBD dataset, NW-UCLA dataset, and the recently proposed NTU-RGBD-120 dataset, shown in Table 4, Table 5, and Table 6 respectively. Our approach exceeds all the previous methods with a notable margin.

Note that the comparison with Directed-GNN is unfair because of the computational cost disparity. Directed-GNN doubles the number of channels in temporal convolution and introduces extra directed graph modules, whose computational cost (127G FLOPs) is nearly double of ours (65G FLOPs) [5]. Nevertheless, we outperform the current state-of-the-art method Directed-GNN at 0.9% on NTU-RGBD X-sub task. On NW-UCLA, we outperform the current state-of-the-art method AGC-LSTM at 2.0%. On NTU-120 RGB+D dataset, we obviously exceed all previously reported performance.

Methods	X-sub	X-view
Lie Group [39]	50.1	52.8
STA-LSTM [36]	73.4	81.2
VA-LSTM [48]	79.2	87.7
ARRN-LSTM [16]	80.7	88.8
Ind-RNN [20]	81.8	88.0
2-Stream 3DCNN [21]	66.8	72.6
TCN [11]	74.3	83.1
ClipCNN+MTLN [9]	79.6	84.8
Synthesized CNN [27]	80.0	87.2
CNN+Motion+Trans [15]	83.2	88.8
ST-GCN [46]	81.5	88.3
Motif+VTDB [44]	84.2	90.2
STGR-GCN [13]	86.9	92.3
AS-GCN [18]	86.8	94.2
Non-local adaptive GCN [34]	88.5	95.1
AGC-LSTM [35]	89.2	95.0
Directed-GNN [33]	89.9	96.1
DC-GCN+ADG (ours)	**90.8**	**96.6**

Table 4. Comparisions of the top-1 accuracy (%) with the state-of-the-art methods on the NTU-RGBD dataset.

[5] Details about the computational complexity are provided in supplement material.

Methods	Year	Top-1
Lie Group [39]	2014	74.2
Actionlet ensemble [41]	2014	76.0
Visualization CNN [28]	2017	86.1
Ensemble TS-LSTM [12]	2017	89.2
2s AGC-LSTM [35]	2019	93.3
DC-GCN+ADG (ours)	-	**95.3**

Table 5. Comparisions of the accuracy (%) with the state-of-the-art methods on the NW-UCLA dataset.

Methods	X-sub	X-setup
Part-Aware LSTM [32]	25.5	26.3
Soft RNN [7]	36.3	44.9
Dynamic Skeleton [6]	50.8	54.7
Spatio-Temporal LSTM [25]	55.7	57.9
Internal Feature Fusion [24]	58.2	60.9
GCA-LSTM [26]	58.3	59.2
Multi-Task Learning Network [9]	58.4	57.9
FSNet [23]	59.9	62.4
Multi CNN + RotClips [10]	62.2	61.8
Pose Evolution Map [29]	64.6	66.9
SkeleMotion [1]	67.7	66.9
DC-GCN+ADG (ours)	**86.5**	**88.1**

Table 6. Comparisions of the top-1 accuracy (%) with the state-of-the-art methods on the NTU-RGBD-120 dataset.

5 Conclusion

In this work, we propose decoupling GCN to boost the graph modeling ability for skeleton-based action recognition. In addition, we propose an attention-guided DropGraph module to effectively relieve the crucial over-fitting problem in GCNs. Both these two contributions introduce zero extra computation, zero extra latency, and zero extra GPU memory cost at deployment. Hence, our approach is not only theoretically efficient but also has well practicality and application prospects. Our approach exceeds the current state-of-the-art method on three datasets: NTU-RGBD, NTU-RGBD-120, and NW-UCLA with even less computation. Since enhancing the effectiveness of the graph modeling and reducing the over-fitting risk are two prevalent problems in GCNs, our approach has potential application value for other GCN tasks, such as recommender systems, traffic analysis, natural language processing, computational chemistry.

Acknowledgement This work was supported in part by the National Natural Science Foundation of China under Grant 61876182 and 61872364, in part by the Jiangsu Leading Technology Basic Research Project BK20192004. This work was partly supported by the Open Projects Program of National Laboratory of Pattern Recognition.

References

1. Caetano, C., Sena, J., Brémond, F., Santos, J.A.d., Schwartz, W.R.: Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. arXiv preprint arXiv:1907.13025 (2019)
2. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
3. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1110–1118 (2015)
4. Fernando, B., Gavves, E., Oramas, J.M., Ghodrati, A., Tuytelaars, T.: Modeling video evolution for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5378–5387 (2015)
5. Ghiasi, G., Lin, T.Y., Le, Q.V.: Dropblock: A regularization method for convolutional networks. In: Advances in Neural Information Processing Systems. pp. 10727–10737 (2018)
6. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5344–5352 (2015)
7. Hu, J.F., Zheng, W.S., Ma, L., Wang, G., Lai, J.H., Zhang, J.: Early action prediction by soft regression. IEEE transactions on pattern analysis and machine intelligence (2018)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
9. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3288–3297 (2017)
10. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: Learning clip representations for skeleton-based 3d action recognition. IEEE Transactions on Image Processing **27**(6), 2842–2855 (2018)
11. Kim, T.S., Reiter, A.: Interpretable 3d human action analysis with temporal convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW). pp. 1623–1631. IEEE (2017)
12. Lee, I., Kim, D., Kang, S., Lee, S.: Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1012–1020 (2017)
13. Li, B., Li, X., Zhang, Z., Wu, F.: Spatio-temporal graph routing for skeleton-based action recognition (2019)
14. Li, B., Dai, Y., Cheng, X., Chen, H., Lin, Y., He, M.: Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. In: 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). pp. 601–604. IEEE (2017)
15. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: 2017 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops, Hong Kong, China, July 10-14, 2017. pp. 597–600 (2017). https://doi.org/10.1109/ICMEW.2017.8026285, `https://doi.org/10.1109/ICMEW.2017.8026285`
16. Li, L., Zheng, W., Zhang, Z., Huang, Y., Wang, L.: Skeleton-based relational modeling for action recognition. CoRR **abs/1805.02556** (2018), `http://arxiv.org/abs/1805.02556`

17. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
18. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3595–3603 (2019)
19. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
20. Li, S., Li, W., Cook, C., Zhu, C., Gao, Y.: Independently recurrent neural network (indrnn): Building a longer and deeper RNN. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 5457–5466 (2018). https://doi.org/10.1109/CVPR.2018.00572, http://openaccess.thecvf.com/content_cvpr_2018/html/Li_Independently_Recurrent_Neural_CVPR_2018_paper.html
21. Liu, H., Tu, J., Liu, M.: Two-stream 3d convolutional neural network for skeleton-based action recognition. arXiv preprint arXiv:1705.08106 (2017)
22. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L., Kot, A.C.: NTU RGB+D 120: A large-scale benchmark for 3d human activity understanding. CoRR **abs/1905.04757** (2019), http://arxiv.org/abs/1905.04757
23. Liu, J., Shahroudy, A., Wang, G., Duan, L.Y., Chichung, A.K.: Skeleton-based online action prediction using scale selection network. IEEE transactions on pattern analysis and machine intelligence (2019)
24. Liu, J., Shahroudy, A., Xu, D., Kot, A.C., Wang, G.: Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE transactions on pattern analysis and machine intelligence **40**(12), 3007–3021 (2017)
25. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: European Conference on Computer Vision. pp. 816–833. Springer (2016)
26. Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C.: Global context-aware attention lstm networks for 3d action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1647–1656 (2017)
27. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition **68**, 346–362 (2017). https://doi.org/10.1016/j.patcog.2017.02.030, https://doi.org/10.1016/j.patcog.2017.02.030
28. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition **68**, 346–362 (2017)
29. Liu, M., Yuan, J.: Recognizing human actions as the evolution of pose estimation maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1159–1168 (2018)
30. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440 (2016)
31. Rong, Y., Huang, W., Xu, T., Huang, J.: Dropedge: Towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations (2020)
32. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1010–1019 (2016)

33. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
34. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
35. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
36. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: Thirty-first AAAI conference on artificial intelligence (2017)
37. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
38. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
39. Veeriah, V., Zhuang, N., Qi, G.J.: Differential recurrent neural networks for action recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 4041–4049 (2015)
40. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 588–595 (2014). https://doi.org/10.1109/CVPR.2014.82, https://doi.org/10.1109/CVPR.2014.82
41. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Learning actionlet ensemble for 3d human action recognition. IEEE transactions on pattern analysis and machine intelligence **36**(5), 914–927 (2013)
42. Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.C.: Cross-view action modeling, learning and recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2649–2656 (2014)
43. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
44. Wen, Y.H., Gao, L., Fu, H., Zhang, F.L., Xia, S.: Graph cnns with motif and variable temporal block for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8989–8996 (2019)
45. Wen, Y., Gao, L., Fu, H., Zhang, F., Xia, S.: Graph cnns with motif and variable temporal block for skeleton-based action recognition. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 8989–8996 (2019), https://aaai.org/ojs/index.php/AAAI/article/view/4929
46. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
47. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)

48. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2117–2126 (2017)
49. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3d human pose regression. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3425–3435. Computer Vision Foundation / IEEE (2019)
50. Zheng, W., Li, L., Zhang, Z., Huang, Y., Wang, L.: Skeleton-based relational modeling for action recognition. arXiv preprint arXiv:1805.02556 (2018)