DGCN: Dynamic Graph Convolutional Network for Efficient Multi-Person Pose Estimation

Zhongwei Qiu[1,2*], **Kai Qiu**[3], **Jianlong Fu**[3], **Dongmei Fu**[1,2]
[1]School of Automation and Electrical Engineering, University of Science and Technology Beijing
[2]Beijing Engineering Research Center of Industrial Spectrum Imaging
[3]Microsoft Research Asia
qiuzhongwei@xs.ustb.edu.cn, {kaqiu, jianf}@microsoft.com, fdm_ustb@ustb.edu.cn

Abstract

Multi-person pose estimation aims to detect human keypoints from images with multiple persons. Bottom-up methods for multi-person pose estimation have attracted extensive attention, owing to the good balance between efficiency and accuracy. Recent bottom-up methods usually follow the principle of keypoints localization and grouping, where relations between keypoints are the keys to group keypoints. These relations spontaneously construct a graph of keypoints, where the edges represent the relations between two nodes (i.e., keypoints). Existing bottom-up methods mainly define relations by empirically picking out edges from this graph, while omitting edges that may contain useful semantic relations. In this paper, we propose a novel Dynamic Graph Convolutional Module (DGCM) to model rich relations in the keypoints graph. Specifically, we take into account all relations (all edges of the graph) and construct dynamic graphs to tolerate large variations of human pose. The DGCM is quite lightweight, which allows it to be stacked like a pyramid architecture and learn structural relations from multi-level features. Our network with single DGCM based on ResNet-50 achieves relative gains of 3.2% and 4.8% over state-of-the-art bottom-up methods on COCO keypoints and MPII dataset, respectively.

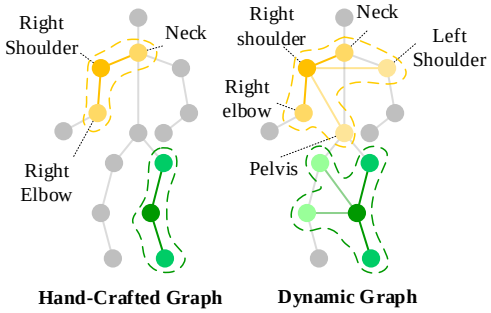

Figure 1: Illustration of the hand-crafted graph and a dynamic graph generated by our model. Left: the hand-crafted graph where each edge represents a physical relationship between two keypoints. Right: a dynamic graph which can model rich relations. Take the right shoulder for example. The hand-crafted graph only models the relations with the neck and right elbow. The dynamic graph can further model the relations with left shoulder and pelvis, which is more robust to occlusion or complex actions. [Best viewed in color].

Introduction

Multi-person human pose estimation aims to recognize human keypoints from images, which usually involve multiple persons. An efficient and accurate human pose estimation approach can benefit extensive real-life applications, including activity recognition (Yan, Xiong, and Lin 2018), human-computer interaction, virtual or augmented reality, AI Coach(Wang et al. 2019), and so on. Although large progress has been seen in recent years, the challenges of large variations in occlusion, truncation, and viewpoints remain.

Two mainstream approaches are prevalent in the field of multi-person pose estimation, including *top-down* (Newell, Yang, and Deng 2016; Chen et al. 2018; Xiao, Wu, and Wei 2018; Sun et al. 2019; Qiu et al. 2019) and *bottom-up* (Cao et al. 2017; Kreiss, Bertoni, and Alahi 2019; Papandreou et al. 2018) manners. The former first detect humans with bounding boxes and then perform single-person pose estimation for each bounding box. The latter directly localize all keypoints from multi-instances and then group keypoints into persons. Bottom-up pose estimation methods attract increasing attention, especially in the industry community, owing to the good balance between efficiency and accuracy.

Since bottom-up methods are box-free, human contextual relations are the keys to identify the keypoints belonging to one instance and distinguish different instances. These relations spontaneously construct a graph, which consists of nodes (keypoints) and edges (relations between keypoints). These edges are extensively used to group keypoints into persons. However, recent bottom-up methods mainly define relations by picking out edges from this graph with hand-

*This work was done when Zhongwei Qiu conducted internship at Microsoft Research Asia.

crafted rules, while the unpicked edges may also contain useful semantic information for pose estimation, as shown in Figure 1.

In this paper, we propose a novel network, named Dynamic Graph Convolutional Network (DGCN), to learn contextual relations of the graph for bottom-up pose estimation. To model rich relations of human keypoints, we construct a graph which contains all the edges between keypoints. Based on the prior that keypoints have strong relations when they are close to each other, we construct a soft graph where the value of each edge is related to the distance of the two keypoints. Note that, this soft graph is obtained by averaging distances between keypoints on the training dataset. Thus this soft graph serves as a static graph. However, the relations of human keypoints dynamically change according to the variations in occlusion, truncation, viewpoints and so on. A static graph is insufficient to model the dynamic relations of human pose. To relieve this problem, we propose to construct dynamic graphs to improve the robustness of networks. Specifically, each element in a dynamic graph conforms to a Bernoulli distribution, where the element at the same location in the soft graph serves as the probability. This dynamic graph changes in each iteration during training, which largely increase the capacity of the network to cover variations of human poses. During inference, DGCM is frozen to produce consistent output. The DGCM is quite lightweight, allowing it to be stacked like a pyramid architecture to further improve performances.

We conduct extensive ablation studies and comparison experiments on two widely-used datasets, including COCO keypoints and MPII, to demonstrate the effectiveness of our DGCN. Compared with state-of-the-art bottom-up methods, our network with single DGCM based on ResNet-50 achieves relative gain of 3.2% and 4.8% on the two datasets.

Related work

In recent years, benefited from the powerful representation of the convolutional neural network, the pose estimation methods based on CNN bring a great process in 2D pose estimation. Compared with traditional methods (Dantone et al. 2013), which rely on hand-craft features and pictorial structures, recent methods (Cao et al. 2017; Papandreou et al. 2018; Xiao, Wu, and Wei 2018; Kreiss, Bertoni, and Alahi 2019; Sun et al. 2019; Li et al. 2019; Moon, Chang, and Lee 2019) extract deep features by convolutional neural networks and decode features into keypoint heatmaps. Good feature representations with structural information are important to recognize human keypoints (Tang and Wu 2019). Some attention-based methods(Ma et al. 2018; Fu, Zheng, and Mei 2017; Qiu et al. 2019) have been used for capturing good pose feature. Since human pose is a kind of graph with strong structural information, recent works (Yan, Xiong, and Lin 2018; Zhao et al. 2019; Ge et al. 2019) build human pose graph neural networks to deal with skeleton-based task.

Multi-Person Pose Estimation

All of these methods based on CNN for multi-person pose estimation can be grouped into *top-down* methods and *bottom-up* methods. The performance of top-down methods relies on the human detector. Bottom-up methods are box-free and thus usually run fast than top-down methods. Therefore, bottom-up methods are widely used in the industry community. The method of (Cao et al. 2017) can run in real-time, which designs a model to learn keypoint heatmaps and part association fields (PAF) simultaneously. It develops a greedy algorithm to group keypoints into persons. Other methods (Papandreou et al. 2018; Kreiss, Bertoni, and Alahi 2019; Newell, Huang, and Deng 2017) design more fine-grained supervisions to learn better heatmaps and PAF.

Graph Convolutional Network

In order to deal with the data with the graph structure, graph neural network (GCN) is introduced in (Gori, Monfardini, and Scarselli 2005; Scarselli et al. 2008; Kipf and Welling 2017). Spectral perspective and spatial perspective are two mainstreams to construct GCN. Spectral analysis are performed in the former methods (Duvenaud et al. 2015; Li et al. 2016; Kipf and Welling 2017). For the spatial domain, the methods of (Bruna et al. 2014; Niepert, Ahmed, and Kutzkov 2016) construct graph CNN filters, which can be applied to the graph nodes and their neighbors. Inspired by the second stream, we construct dynamic graph convolutional networks to learn relations of human keypoints.

Approach

Bottom-up pose estimation methods try to localize keypoints and group keypoints into persons using the relations between keypoints. Recent works mainly rely on modeling limb relations between keypoints pairs to group keypoints, while the body relations among keypoints graph are neglected. The fact that human body keypoints construct a graph naturally motivates us to design novel graph convolutional network (GCN) to model body relations among keypoints graph.

In this section, we first introduce the whole pipeline for bottom-up pose estimation. Then, we describe how to leverage GCN to model relations among keypoints graph. Last, we propose DGCN which constructs dynamic keypoints graphs based on the spatial relations of keypoints to tolerate human pose variations.

Bottom-up Pose Estimation Pipeline

Bottom-up pose estimation methods try to learn two kinds of heatmaps from the deep neural network, including keypoint heatmaps and relation heatmaps. From the keypoint heatmaps, keypoints can be localized by searching for the local peaks. Using the relation heatmaps (usually in the form of limb relation), keypoints can be grouped into persons. In recent years, various ideas are explored to optimize the supervisions for heatmaps and grouping strategies.

We follow the state-of-the-art bottom-up method (Kreiss, Bertoni, and Alahi 2019) to construct supervisions for training keypoint heatmaps and relation heatmaps (called PIF and PAF in this paper). Specifically, keypoint heatmaps consist of confidence maps H_c and offsets maps $\{H_x, H_y\}$, while relation heatmaps consist of limb confidence maps A_c and limb offset maps $\{A_x, A_y\}$. Binary cross-entropy loss is

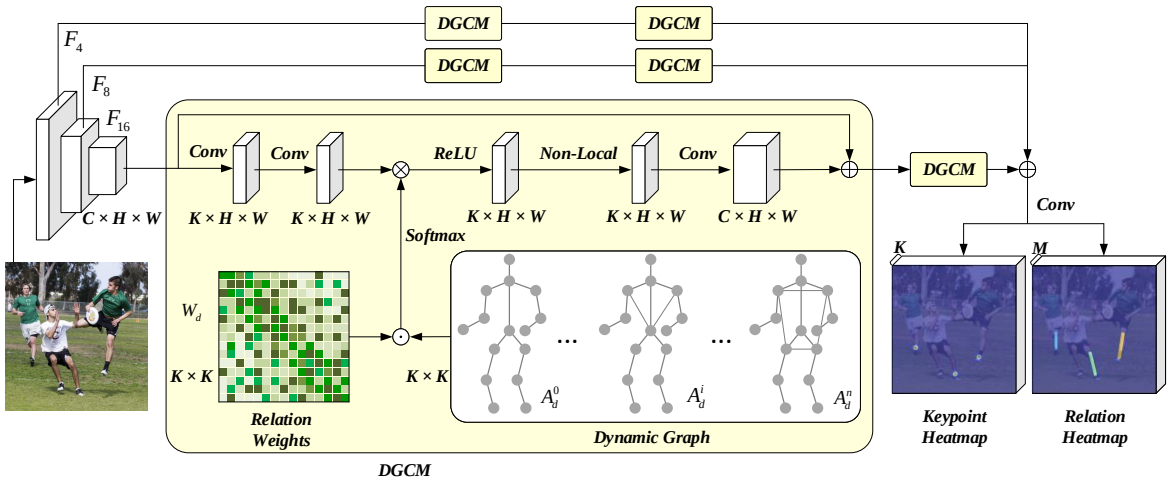

Figure 2: The architecture of pyramid DGCN. Given an image, we extract deep features F_s (s denotes stride, $s = 4, 8, 16$) from the backbone (e.g. ResNet). For a feature with the shape of $C \times H \times W$, DGCM first reduce the number of channels to K (K denotes the number of predefined keypoints), and then construct a dynamic graph A_d with the shape of $K \times K$ to model relations between these K channels. W_d is learnable relation weights with the shape of $K \times K$. The outputs of DGCM are decoded into keypoint heatmaps and relation heatmaps. $Conv$ denotes the convolution layer. $\odot$ denotes element-wise multiplication. $\otimes$ denotes matrix multiplication. $\oplus$ denotes element-wise sum. We randomly select one of A_d^0 ... A_d^n in each iteration during training.

used for learning confidence maps and $Smooth - L_1$ loss is used for learning offsets maps.

$$\ell = \ell_H + \ell_A \tag{1}$$

$$\ell_H = \sum_k^K (\alpha \cdot \phi(H_c - H_c^*) + \theta \cdot \varphi(H_{xy} - H_{xy}^*)) \tag{2}$$

$$\ell_A = \sum_m^M (\beta \cdot \phi(A_c - A_c^*) + \delta \cdot \varphi(A_{xy} - A_{xy}^*)) \tag{3}$$

where ℓ is total loss. ℓ_H and ℓ_A are keypoint heatmaps loss and relation heatmaps loss. ϕ and φ are binary cross-entropy loss and $Smooth - L_1$ loss, respectively. $\alpha, \theta, \beta, \delta$ denote loss weights. K and M denote the number of human keypoints and human defined relations, respectively. $H_c^*, H_{xy}^*, A_c^*, A_{xy}^*$ are groundtruth.

The grouping strategy follows the greedy decoding idea to group keypoints into persons, which is not the focal point of this paper, for more details refer to (Kreiss, Bertoni, and Alahi 2019). The framework of our DGCN is shown in Figure 2.

GCN for Keypoints Graph Modeling

The fact that human keypoints construct a graph naturally motivates us to design novel graph convolutional network (GCN) to model body relations among keypoints graph. K human keypoints construct a graph, which contains K vertexes and K^2 edges. Each edge models the relation between two keypoints. These K^2 edges can be formed to an adjacency matrix. We use this adjacency matrix to model relations over keypoint features.

We first introduce a basic GCN for modeling keypoints relations, which is also described in (Zhao et al. 2019). The keypoints graph $G = (V, E)$ consists of the keypoints set $V = \{v_i | i = 1, ..., K\}$ and limbs set $E = \{e_i | i = 1, ..., M\}$, where M denotes the number of hand-crafted limbs. Let X_i^l denote the representation of keypoint v_i in the lth layer. We define $A \in [0, 1]^{k \times k}$ as the adjacency matrix of graph G, where $a_{ij} = 1$ when the ith keypoint has connection with jth keypoint, and $a_{ii} = 1$ for all i. Then, the graph convolution operation can be formulated as

$$X^{l+1} = \sigma(W X^l \widetilde{A}) \tag{4}$$

where $\widetilde{A}$ is symmetrically normalized from A. σ denotes non-linear function (e.g. ReLU). To reduce computation cost, we define W as convolution with kernel size of 1.

Dynamic Graph Convolutional Network

The basic GCN is able to learn relations on the human-defined edges (where $a_{ij} = 1$) of the keypoints graph (here the human-defined adjacency matrix is denoted as A_h), while it may miss important information on the undefined edges (where $a_{ij} = 0$). To solve this problem, we propose to use a soft adjacency matrix where a_{ij} is related to the spatial distance of two keypoints. Specifically, this soft adjacency matrix exploits the prior that keypoints are closer to each other have stronger relations. Experiments in the next section show the superiority of this soft adjacency matrix over basic GCN.

Since the relations between human keypoints change dynamically according to the viewpoints, occlusion, and truncation, we propose DGCN to further improve the capacity

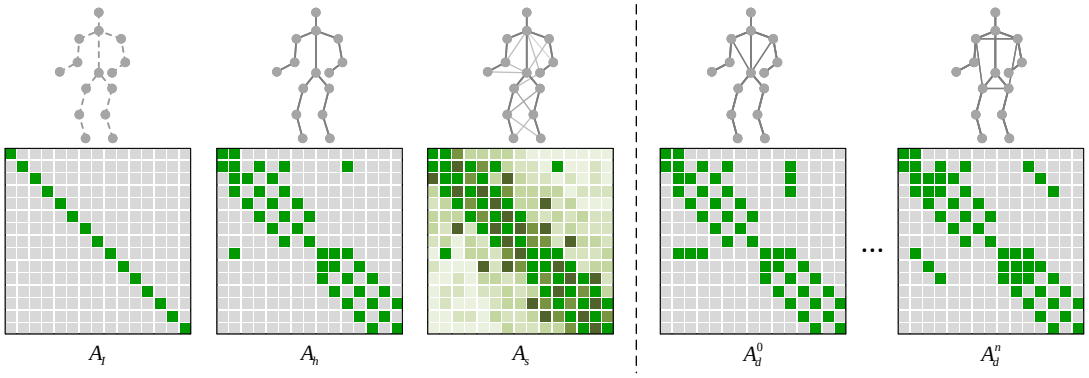

Figure 3: Different adjacency matrixes for GCN. The shape of A is $K \times K$, which represents the relation among keypoints. A_I is diagonal matrix. A_h is hand-crafted adjacency matrix. A_s is soft adjacency matrix. A_d is dynamic adjacency matrix generated by DGCM.

of our model to tolerate human pose variations. Specifically, we adopt a sampling strategy to construct dynamic graph based on the soft adjacency matrix. We construct dynamic graph during training and freeze it during inference, just like Dropout (Srivastava et al. 2014).

Soft Adjacency Matrix The distance matrix M_d is derived from the training dataset. The distance M_d^{ij} between two keypoints V^i and V^j is

$$M_d^{ij} = \frac{1}{N}\sum_{n=0}^{N}\frac{1}{s_n}\|V^i - V^j\|_2 \tag{5}$$

where s_n denotes the scale of the person and N denotes the numbers of all persons in training dataset. The diagonal values of M_d are 0.

The soft adjacency matrix A_s is obtained by a $softmax$ function over distance matrix M_d with a scale parameter γ.

$$A_s^{ij} = \sigma(\gamma\frac{1}{M_d^{ij}}) \tag{6}$$

where $softmax$ function σ is applied on the rows (diagonal values are ignored). Then we set $A_s^{ii} = 1$ for all diagonal elements.

Dynamic Adjacency Matrix Dynamic adjacency matrix A_d is constructed based on the soft adjacency matrix A_s. Specifically, each element A_d^{ij} conforms to a Bernoulli distribution, where the corresponding soft adjacency value A_s^{ij} serves as the probability

$$A_d^{ij} \sim B(x, A_s^{ij}) \tag{7}$$

where B is Bernoulli distribution. During training, A_d dynamically changes in each iteration, according to the Bernoulli distribution. For inference, $A_d^{ij} = 1$ for all elements.

We also study the influence of different adjacency matrix. As shown in Figure 3, different adjacency represents different pose graph.

Table 1: The Results on COCO Keypoints val2017. The input size of an image is 641×641. Backbone is ResNet-50. The baseline is the model without adding GCN modules (Kreiss, Bertoni, and Alahi 2019). I means unit diagonal matrix and use in equation 4, which is a simple version of the GCN model. A_h means a fixed human keypoints adjacency matrix constructed by prior knowledge and use in equation 8. A_s and A_d are introduced in section methods. We use F_{16} only.

Method	baseline	GCN-A_I	GCN-A_h	GCN-A_s	DGCN-A_d
AP	0.626	0.636	0.639	0.641	0.646

Dynamic Graph Convolution Module After obtaining the dynamic adjacency matrix A_d, we recap the GCN module from equation 4 as dynamic version:

$$X^{l+1} = \sigma(WX^l\beta(W_d \odot A)) \tag{8}$$

where $W_d \in \mathcal{R}^{K\times K}$ is a learnable weights matrix accompanied with dynamic adjacency matrix A. $\odot$ is element-wise multiplication. β is a $softmax$ operation performed on each row of $(W_d \odot A)$. σ is a ReLU layer. During training, W_d is updated by the supervision of the loss function and learn to model the weight of each limb relation.

Equation 8 is designed to learn relations between keypoints. We also follow (Zhao et al. 2019) to add a non-local layer to learn spatial relations from features. For simplicity, this module is named as DGCM. More details are shown in Figure 2.

Pyramid DGCN Since the scales of instances are different, multi-scale features are useful for improving the adaptive ability of the network. To further explore the capacity of DGCN, we design and implement the pyramid DGCN.

As shown in Figure 2, we extract features from multiple stages of the ResNet backbone. Let F_s denotes the feature extracted from the stage that has a stride of s based on the input image. We use three features from the ResNet backbone

Table 2: The performance of DGCN based on ResNet-50 on COCO Keypoints val2017 with different weights scale parameter γ. We use A_s as adjacency matrix. We use F_{16} only.

γ	0.01	0.1	0.5	1	5	100
AP	0.638	0.639	0.641	0.641	0.641	0.639

Table 3: The results of pyramid DGCN on COCO Keypoints val2017. Input size of an image is 641×641. DGCN-50 denotes the DGCN model is based on ResNet-50. F_s denotes the DGCN head(one DGCN head consists of two DGCMs) used in model.

Method	A	DGCN-Head	AP
baseline	-	-	0.626
DGCN-50	A_d	F_{16}	0.646
DGCN-50	A_d	F_{16} & F_8	0.651
DGCN-50	A_d	F_{16} & F_8 & F_4	0.652

to form a pyramid, including F_4, F_8 and F_{16}. Each feature goes through a DGCM and the outputs are downsampled to the same size and combined by an addition operation:

$$F_g = \sum_{s \in \{4,8,16\}} \psi(F_s) \tag{9}$$

where ψ denotes the DGCM. The output feature of pyramid DGCN is used for predicting keypoint heatmaps and limb heatmaps.

Experiments

In this section, we introduce the details of implementing DGCN and experiments. We use two pose estimation datasets: MS COCO and MPII. We conduct ablation studies on COCO and report comparison results on two datasets.

Implement details

As Figure 2 shown, the channels of deep features F_s are reduced to K which represents the number of keypoints of one person. Then, intermediate supervisions of keypoints confidence maps are used after reducing channels. The weight of intermediate supervision is $\frac{\alpha}{N}$ for training and N represents the number of DGCM. α, θ, β and δ equal 30, 2, 50 and 3, respectively. We set γ as 0.5 for all experiments, except for the ablation experiments on γ. ResNet pre-trained on ImageNet is used. During training, we employ SGD for optimization with an initial learning rate of 0.001. We freeze the weights of ResNet in the first epoch and the total epoch is 100. Two NVIDIA Tesla V100 GPUs are used and batch size is 8.

We init A_h for graph convolutional network following the human skeleton knowledge in the basic GCN version. A_h is fixed for testing. However, A_d is dynamic in training. Therefore, we set $A_d^{ij} = 1$ for testing.

Dataset & Evaluation

We conduct experiments on MS COCO (Lin et al. 2014), MPII (Andriluka et al. 2014), and CrowdPose (Li et al. 2019) datasets.

Table 4: Comparison of model params with the baseline model of **Pifpaf** (Kreiss, Bertoni, and Alahi 2019) with different backbones.

Method	DGCN heads	params (MB)	AP
baseline-50	-	96	0.626
DGCN-50	1	102	0.646
DGCN-50	2	121	0.651
DGCN-50	3	127	0.652
baseline-101	-	169	0.657
DGCN-101	1	174	0.673
baseline-152	-	229	0.674
DGCN-152	1	234	0.688

Table 5: Results of **Pifpaf** and DGCN with different backbones on COCO and crowdpose(Li et al. 2019) dataset.

Method	DGCN heads	Dataset	AP
baseline-ResNext50	-	COCO	0.638
DGCN-ResNext50	1		0.651
baseline-DensNet121	-		0.618
DGCN-DensNet121	1		0.636
baseline-ResNet50	-	CrowdPose	0.563
DGCN-ResNet50	1		0.591

COCO is a large database with more than 200k images. More than 150k human instances are annotated in the train and validation dataset. The annotation of the COCO dataset contains 17 keypoints for a person, and the invisible keypoints will be annotated specifically. The evaluation metric on the COCO dataset is mAP based on object keypoints similarity(OKS). We evaluate on COCO validation dataset and test-dev dataset.

MPII dataset includes over 25k images. More than 40k human instances are annotated and each person is annotated with 16 keypoints. The evaluation metric is PCKh which calculates the precision of correct keypoints with respect to head.

CrowdPose consists of 20k images, containing about 80k persons. Each person is annotated with 14 keypoints. CrowdPose dataset follows the evaulation metric of COCO, but more persons in an same image, which is more difficult.

Ablation Study

We demonstrate the effectiveness of DGCN on the COCO keypoints dataset. And we study the impact of different soft adjacency matrix A_s. Then, based on DGCN, we also show the performance of the pyramid DGCN on COCO keypoints dataset.

GCN & DGCN Our baseline is the bottom-up method (Kreiss, Bertoni, and Alahi 2019) for 2D multi-person pose estimation with ResNet backbone. Two heads are connected on backbone for generating keypoints confidence maps and keypoints association maps. This method has the state-of-the-art performance (mAP of 62.6 based ResNet-50 backbone) in bottom-up methods on COCO keypoints dataset.

Table 6: Comparison with stat-of-the-art bottom-up methods on COCO Keypoints val2017. Only one DGCN head is used on the F_{16}. Results of **Pifpaf** are cited from (Kreiss, Bertoni, and Alahi 2019). Results of **Personlab** are cited from (Papandreou et al. 2018). They just provide the results on ResNet-101 in those paper.

Method	Backbone	Input Size	AP	$AP^{.50}$	$AP^{.75}$	AP^{M}	AP^{L}	AR	$AR^{.50}$	$AR^{.75}$	AR^{M}	AR^{L}
Pifpaf	R-50	641	0.626	0.851	0.687	0.599	0.674	0.686	0.884	0.741	0.639	0.751
DGCN-50(ours)			**0.646**	**0.853**	**0.708**	**0.615**	**0.695**	**0.702**	**0.886**	**0.756**	**0.656**	**0.765**
Personlab	R-101	601	0.541	0.764	0.577	0.406	0.733	0.577	0.787	0.613	0.435	0.774
Personlab		1001	0.646	0.854	0.698	0.576	**0.753**	0.684	0.873	0.735	0.608	0.793
Personlab		1401	0.665	0.862	0.719	0.623	0.732	0.707	0.887	0.757	0.656	0.779
Pifpaf		641	0.657	0.866	0.719	0.619	0.718	0.712	**0.895**	0.768	0.660	0.785
DGCN-101(ours)		641	**0.673**	**0.867**	**0.741**	**0.638**	0.727	**0.722**	0.894	**0.781**	**0.676**	**0.788**
Pifpaf	R-152	641	0.674	0.869	0.738	0.631	0.741	0.726	0.898	0.781	0.672	0.800
DGCN-152(ours)		641	**0.688**	**0.875**	**0.755**	**0.653**	**0.744**	**0.737**	**0.902**	**0.794**	**0.690**	**0.802**

Table 7: Results on of our single DGCN model on COCO Keypoints test-dev2017. Input size of image is 641×641. Only one DGCN head is used on F_{16}. The results of other methods are cited from (Cao et al. 2017), (Newell, Huang, and Deng 2017), (Papandreou et al. 2018) and (Kreiss, Bertoni, and Alahi 2019), respectively.

Method	AP	$AP^{.50}$	$AP^{.75}$	AP^{M}	AP^{L}	AR	$AR^{.50}$	$AR^{.75}$	AR^{M}	AR^{L}
CMU-Pose	0.618	0.849	0.675	0.571	0.682	0.665	0.872	0.718	0.606	0.746
AE	0.630	0.857	0.689	0.580	0.704	-	-	-	-	-
AE (refine)	0.655	0.868	0.723	0.606	0.726	0.702	0.895	0.760	0.646	0.781
Personlab	0.665	**0.880**	0.726	0.624	0.723	0.710	0.903	0.766	0.661	0.777
Pifpaf	0.667	0.878	0.736	0.624	0.729	0.722	0.909	0.783	0.664	0.800
DGCN-152(ours)	**0.674**	**0.880**	**0.744**	**0.636**	**0.730**	**0.732**	**0.913**	**0.792**	**0.680**	**0.802**

Following the settings of the experiment of baseline, we conduct ablation studies on the COCO validation dataset.

First, we add a simple GCN head on features F_{16} with $\widetilde{A} = I$ following the equation 4. As shown in Table 1 method GCN-A_I, the simple GCN modules lead to a relative improvement of 1.6%, which verifies the effectiveness of the GCN model.

Second, we change GCN modules as equation 8, and set A as A_h which is generated by human skeleton knowledge and freeze A_h. Compared with method GCN-A_I, method GCN-A_h brings more potential keypoints relations by keypoints adjacency matrix A_h. As shown in Table 1 method DGCN-A_h, the GCN model with a fixed keypoints relation matrix A_h leads to a relative improvement of 2.1% based on ResNet-50.

Third, as mentioned in the last section, its difficult to decide which keypoint should connect with the other keypoints, which results in the different keypoints adjacency matrix A_h from different researchers. Therefore, we design dynamic GCN to handle this problem. We conduct an ablation study about GCN with a soft adjacency matrix. Following the equation 8, we set $A = A_s$ (A_s come from equation 6). A_s^{ij} represents the probability of keypoint i related to keypoints j. As shown in Table 1 method GCN-A_s, the GCN model with soft keypoints adjacency matrix A_s leads to a relative improvement of 2.4%. Then, we change the fixed soft keypoints adjacency matrix A_s to dynamic adjacency matrix A_d, which leads to a relative improvement of 3.2%.

In addition, there is a parameter γ for generating keypoints relation probability matrix A_s, which controls the scales of relation weights between keypoints and keypoints. We also study the impact of different γ. As shown in Table 2, there is a little difference in performance with different γ. We think that the learning weights matrix W_d counteract the influence of different γ. But we find that a small or a large γ will influence the stability of the training model from experiments. Therefore, we set $\gamma = 0.5$ for other experiments.

Pyramid DGCN Multi-scale features are useful for multi-person pose estimation in bottom-up methods, because of the different scales of people in an image. To explore the capacity of the proposed DGCN models, we design a pyramid DGCN model to learn multi-scale graph features. According to the different feature map size in F_s (s represents stride), we build a graph feature pyramid network based on ResNet-50. We firstly only add one DGCN head on F_{16}, then obtain graph features F_{g16}, which further used to generate keypoint heatmaps and relation heatmaps. The DGCN-50 model with one DGCN head obtains an mAP of 64.6 (shown in Table 3). After adding graph heads on F_{16} and F_8, we sum the features F_{g16} and F_{g8}. Then we just decode the sum of multi-scale features, which leads to a relative improvement of 4.0% compared with baseline. Finally, we add three graph heads on F_{16}, F_8 and F_4, respectively. Then, we also decode the sum of these multi-scale features. We find that there are a little improvement from 2 DGCN heads to 3 DGCN heads. The performance of the pyramid DGCN is saturated with 3 DGCN heads.

In addition, we make a comparison on COCO and CrowdPose dataset with different backbones. As shown in table 5, DGCN outperform the baseline methods on ResNext and DensNet backbones. On the more difficult CrowdPose dataset, DGCN leads a relative improvement of 5.0%.

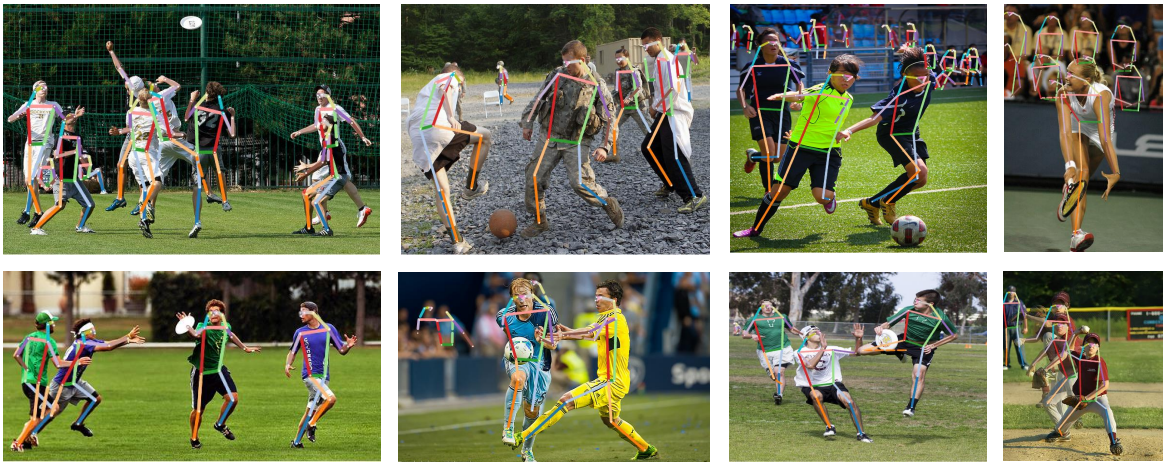

Figure 4: Visualization of the results produced by our DGCN. It shows that DGCN performs well even for challenging cases.

Table 8: Comparison with bottom-up methods on MPII. The results of Joint-Graph, Arttrack, CMU-Pose, RMPE and AE are cited from (Levinkov et al. 2017; Insafutdinov et al. 2017; Cao et al. 2017; Fang et al. 2017; Newell, Huang, and Deng 2017). Our DGCN-50 has one DGCN head on F_{16}.

Method	*Head*	*Shoulder*	*Elbow*	*Wrist*	*Hip*	*Knee*	*Ankle*	*Mean*
Joint-Graph	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6
Arttrack	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3
CMU-Pose	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6
RMPE	88.4	86.5	78.6	70.4	74.4	73.0	**65.8**	76.7
AE	92.1	89.3	78.9	69.8	76.2	71.6	64.7	77.5
DGCN-50(ours)	**95.6**	**92.5**	**83.1**	**76.5**	**81.5**	**73.1**	65.1	**81.2**

Comparison of Parameters

Compared with the state-of-the-art method (Kreiss, Bertoni, and Alahi 2019) in bottom-up methods, as shown in Table 4, we get a relative improvement of 3.2% AP with adding one DGCN head based on ResNet-50, which just increases $6MB$ params. From 1 DGCN head to 2 DGCN head, there are more about $20MB$ params. The reason is that there are more downsampling layers from F_8 to F_{16}. In summary, our DGCN model gains the stat-of-the-art results with small extra params.

Comparison Experiments on COCO Dataset

COCO Keypoints Validation On the COCO val2017 dataset, we follow the standard experiment settings as the-state-of-arts approaches (Kreiss, Bertoni, and Alahi 2019), we report our results with a single scale graph GCN model based on ResNet-50, ResNet-101 and ResNet-152, respectively. As Table 6 shown, compared with Personlab (Papandreou et al. 2018), we outperform their best results with large input size. Compared the Pifpaf (Kreiss, Bertoni, and Alahi 2019) (They didn't provide the results in different metrics scales, so just comparing on the final AP), we outperform their results with relative 3.2%, 2.4% and 2.1% AP based on the different backbone. All the results of our DGCN are gained with only one DGCN head.

COCO Keypoints Test-dev We also report our results on the COCO keypoints test-dev dataset. We conduct a comparison with other bottom-up methods for multi-person pose estimation. As shown in Table 7, compared with the-stat-of-art method (Kreiss, Bertoni, and Alahi 2019), we gain relative 1% overall AP increasing. Visualization results are shown in Figure 4.

Comparison Experiments on MPII Dataset

As shown in Table 8, the PCKh on symmetric keypoints (such as shoulders, elbows ...) is the average of left keypoints and right keypoints. Our DGCN also achieves the state-of-the-art performance in bottom-up methods on the MPII dataset.

Conclusion

In this paper, we present a novel DGCN for 2D multi-person pose estimation. DGCN aims to learn rich relations between human keypoints and tolerate large variations of human pose. Extensive ablation studies and comparison experiments on two widely-used datasets demonstrate the effectiveness of DGCN. We also notice some limitations of this work. First, DGCN is only used for learning relations from features in this paper, while it should also work for grouping keypoints into persons. Second, a keypoints graph related to

human action may work better than the current DGCN. We leave these for future exploration.

Acknowledgments

This work is supported by the Fundamental Research Funds for the China Central Universities of USTB (No. FRF-BD-19-002A) and Beijing Key Discipline Development Program of Beijing Municipal Commission (No. XK100080537).

References

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B. 2014. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 3686–3693.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 7291–7299.

Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; and Sun, J. 2018. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 7103–7112.

Dantone, M.; Gall, J.; Leistner, C.; and Van Gool, L. 2013. Human pose estimation using body parts dependent joint regressors. In *CVPR*, 3041–3048.

Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; and Adams, R. P. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, 2224–2232.

Fang, H.-S.; Xie, S.; Tai, Y.-W.; and Lu, C. 2017. Rmpe: Regional multi-person pose estimation. In *CVPR*, 2334–2343.

Fu, J.; Zheng, H.; and Mei, T. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, 4438–4446.

Ge, L.; Ren, Z.; Li, Y.; Xue, Z.; Wang, Y.; Cai, J.; and Yuan, J. 2019. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 10833–10842.

Gori, M.; Monfardini, G.; and Scarselli, F. 2005. A new model for learning in graph domains. In *Proceedings of IEEE International Joint Conference on Neural Networks*, volume 2, 729–734. IEEE.

Insafutdinov, E.; Andriluka, M.; Pishchulin, L.; Tang, S.; Levinkov, E.; Andres, B.; and Schiele, B. 2017. Arttrack: Articulated multi-person tracking in the wild. In *CVPR*, 6457–6465.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kreiss, S.; Bertoni, L.; and Alahi, A. 2019. Pifpaf: Composite fields for human pose estimation. In *CVPR*, 11977–11986.

Levinkov, E.; Uhrig, J.; Tang, S.; Omran, M.; Insafutdinov, E.; Kirillov, A.; Rother, C.; Brox, T.; Schiele, B.; and Andres, B. 2017. Joint graph decomposition & node labeling: Problem, algorithms, applications. In *CVPR*, 6012–6020.

Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. 2016. Gated graph sequence neural networks. In *ICLR*.

Li, J.; Wang, C.; Zhu, H.; Mao, Y.; Fang, H.-S.; and Lu, C. 2019. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *CVPR*, 10863–10872.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Ma, S.; Fu, J.; Wen Chen, C.; and Mei, T. 2018. Da-gan: Instance-level image translation by deep attention generative adversarial networks. In *CVPR*, 5657–5666.

Moon, G.; Chang, J. Y.; and Lee, K. M. 2019. Posefix: Model-agnostic general human pose refinement network. In *CVPR*, 7773–7781.

Newell, A.; Huang, Z.; and Deng, J. 2017. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, 2277–2287.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In *ECCV*, 483–499. Springer.

Niepert, M.; Ahmed, M.; and Kutzkov, K. 2016. Learning convolutional neural networks for graphs. In *ICML*, 2014–2023.

Papandreou, G.; Zhu, T.; Chen, L.-C.; Gidaris, S.; Tompson, J.; and Murphy, K. 2018. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 269–286.

Qiu, Z.; Qiu, K.; Fu, J.; and Fu, D. 2019. Learning recurrent structure-guided attention network for multi-person pose estimation. In *ICME*, 418–423. IEEE.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. volume 20, 61–80. IEEE.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 5693–5703.

Tang, W., and Wu, Y. 2019. Does learning specific features for related parts help human pose estimation? In *CVPR*, 1107–1116.

Wang, J.; Qiu, K.; Peng, H.; Fu, J.; and Zhu, J. 2019. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In *ACM MM*, 374–382.

Xiao, B.; Wu, H.; and Wei, Y. 2018. Simple baselines for human pose estimation and tracking. In *ECCV*, 466–481.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; and Metaxas, D. N. 2019. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, 3425–3435.