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

Zhongwei Qiu,1,2+ Kai Qiu,3 Jianlong Fu,3 Dongmei Fu1,2
1School of Automation and Electrical Engineering, University of Science and Technology Beijing
2Beijing Engineering Research Center of Industrial Spectrum Imaging
3Microsoft 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 Convolution 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.

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-
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, DGCN is frozen to produce consistent output. The DGCN 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 $H_z$ and offsets maps $\{H_x, H_y\}$, while relation heatmaps consist of limb confidence maps $A_z$ and limb offset maps $\{A_x, A_y\}$. Binary cross-entropy loss is
used for learning confidence maps and $\text{Smooth} - L_1$ loss is used for learning offsets maps.

\[
\ell = \ell_H + \ell_A
\]  

\[
\ell_H = \sum_k^K (\alpha \cdot \phi(H_c - H^*_c) + \theta \cdot \varphi(H_{xy} - H^*_{xy}))
\]  

\[
\ell_A = \sum_m^M (\beta \cdot \phi(A_c - A^*_c) + \delta \cdot \varphi(A_{xy} - A^*_{xy}))
\]

where $\ell$ is total loss, $\ell_H$ and $\ell_A$ are keypoint heatmaps loss and relation heatmaps loss, $\phi$ and $\varphi$ are binary cross-entropy loss and $\text{Smooth} - L_1$ 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 $K$ vertices 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 keypoint 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 $E = \{e_{ij} | i = 1, ..., M\}$, where $M$ denotes the number of hand-crafted limbs. Let $X^l_i$ denote the representation of keypoint $v_i$ in the $l$th layer. We define $A \in [0, 1]^{k \times k}$ as the adjacency matrix of graph $G$, where $a_{ij} = 1$ when the $i$th keypoint has connection with $j$th keypoint, and $a_{ii} = 1$ for all $i$. Then, the graph convolution operation can be formulated as

\[
X^{l+1} = \sigma(W X^l \tilde{A})
\]

where $\tilde{A}$ is symmetrically normalized from $A$. $\sigma$ 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.
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

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

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 $\gamma$.

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

where softmax function $\sigma$ 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

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

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.

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

$$X^{i+1} = \sigma(W X^i \beta(W_d \odot A))$$

where $W_d \in \mathbb{R}^{K \times K}$ is a learnable weights matrix accompanied with dynamic adjacency matrix $A$. $\odot$ is element-wise multiplication. $\beta$ is a *softmax* operation performed on each row of $(W_d \odot A)$. $\sigma$ 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 $F_x$ 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 \( \gamma \). We use \( A_s \) as adjacency matrix. We use \( F_{16} \) only.

<table>
<thead>
<tr>
<th>( \gamma )</th>
<th>0.01</th>
<th>0.1</th>
<th>0.5</th>
<th>1</th>
<th>5</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.638</td>
<td>0.639</td>
<td>0.641</td>
<td>0.641</td>
<td>0.641</td>
<td>0.639</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>A</th>
<th>DGCN-Head</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td></td>
<td>-</td>
<td>0.626</td>
</tr>
<tr>
<td>DGCN-50</td>
<td>( A_d )</td>
<td>( F_{16} )</td>
<td>0.646</td>
</tr>
<tr>
<td>DGCN-50</td>
<td>( A_d )</td>
<td>( F_{16} &amp; F_8 )</td>
<td>0.651</td>
</tr>
<tr>
<td>DGCN-50</td>
<td>( A_d )</td>
<td>( F_{16} &amp; F_8 &amp; F_4 )</td>
<td>0.652</td>
</tr>
</tbody>
</table>

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

\[
F_g = \sum_{s \in \{4, 8, 16\}} \psi(F_s)
\]

where \( \psi \) 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 \( \frac{\gamma}{N} \) for training and \( N \) represents the number of DGCM. \( \alpha, \theta, \beta \) and \( \delta \) equal 30, 2, 50 and 3, respectively. We set \( \gamma \) as 0.5 for all experiments, except for the ablation experiments on \( \gamma \). 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 \( A_b \) 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 \( A_d^{12} = 1 \) 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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>$A_P$</th>
<th>$A_{P,30}$</th>
<th>$A_{P,50}$</th>
<th>$A_{P,75}$</th>
<th>$A_{P,L}$</th>
<th>$A_R$</th>
<th>$A_{R,30}$</th>
<th>$A_{R,50}$</th>
<th>$A_{R,L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PitPaf</td>
<td>DGCN-50(ours)</td>
<td>641</td>
<td>0.626</td>
<td>0.851</td>
<td>0.687</td>
<td>0.599</td>
<td>0.674</td>
<td>0.686</td>
<td>0.884</td>
<td>0.741</td>
<td>0.639</td>
</tr>
<tr>
<td>Personlab</td>
<td>DGCN-50(ours)</td>
<td>601</td>
<td>0.646</td>
<td>0.853</td>
<td>0.708</td>
<td>0.615</td>
<td>0.695</td>
<td>0.702</td>
<td>0.886</td>
<td>0.756</td>
<td>0.656</td>
</tr>
<tr>
<td>Personlab</td>
<td>DGCN-101(ours)</td>
<td>1001</td>
<td>0.646</td>
<td>0.854</td>
<td>0.698</td>
<td>0.576</td>
<td>0.753</td>
<td>0.684</td>
<td>0.873</td>
<td>0.735</td>
<td>0.608</td>
</tr>
<tr>
<td>Personlab</td>
<td>DGCN-101(ours)</td>
<td>1401</td>
<td>0.665</td>
<td>0.862</td>
<td>0.719</td>
<td>0.619</td>
<td>0.718</td>
<td>0.712</td>
<td>0.895</td>
<td>0.768</td>
<td>0.660</td>
</tr>
<tr>
<td>PitPaf</td>
<td>DGCN-152(ours)</td>
<td>R-152</td>
<td>0.673</td>
<td>0.867</td>
<td>0.741</td>
<td>0.638</td>
<td>0.727</td>
<td>0.722</td>
<td>0.894</td>
<td>0.781</td>
<td>0.676</td>
</tr>
<tr>
<td>PitPaf</td>
<td>DGCN-152(ours)</td>
<td>R-152</td>
<td>0.688</td>
<td>0.875</td>
<td>0.755</td>
<td>0.653</td>
<td>0.744</td>
<td>0.737</td>
<td>0.902</td>
<td>0.794</td>
<td>0.690</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>$A_P$</th>
<th>$A_{P,30}$</th>
<th>$A_{P,50}$</th>
<th>$A_{P,75}$</th>
<th>$A_{P,L}$</th>
<th>$A_R$</th>
<th>$A_{R,30}$</th>
<th>$A_{R,50}$</th>
<th>$A_{R,L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU-Pose</td>
<td>0.618</td>
<td>0.849</td>
<td>0.675</td>
<td>0.571</td>
<td>0.682</td>
<td>0.665</td>
<td>0.872</td>
<td>0.718</td>
<td>0.606</td>
</tr>
<tr>
<td>AE</td>
<td>0.630</td>
<td>0.857</td>
<td>0.689</td>
<td>0.580</td>
<td>0.704</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AE (refine)</td>
<td>0.655</td>
<td>0.868</td>
<td>0.723</td>
<td>0.606</td>
<td>0.726</td>
<td>0.702</td>
<td>0.895</td>
<td>0.760</td>
<td>0.646</td>
</tr>
<tr>
<td>Personlab</td>
<td>0.665</td>
<td>0.880</td>
<td>0.726</td>
<td>0.624</td>
<td>0.723</td>
<td>0.710</td>
<td>0.903</td>
<td>0.766</td>
<td>0.661</td>
</tr>
<tr>
<td>PitPaf</td>
<td>0.667</td>
<td>0.878</td>
<td>0.736</td>
<td>0.624</td>
<td>0.729</td>
<td>0.722</td>
<td>0.909</td>
<td>0.783</td>
<td>0.664</td>
</tr>
<tr>
<td>DGCN-152(ours)</td>
<td>0.674</td>
<td>0.880</td>
<td>0.744</td>
<td>0.636</td>
<td>0.730</td>
<td>0.732</td>
<td>0.913</td>
<td>0.792</td>
<td>0.680</td>
</tr>
</tbody>
</table>

First, we add a simple GCN head on features $F_{16}$ with $A = I$ following the equation 4. As shown in Table 1 method GCN-$A_1$, 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 $A_1$, which is generated by human skeleton knowledge and freeze $A_h$. Compared with method GCN-$A_1$, method GCN-$A_2$ brings more potential keypoints relations by keypoints adjacency matrix $A_2$. As shown in Table 1 method GCN-$A_2$, the GCN module with a fixed keypoints relation matrix $A_1$ leads to a relative improvement of 2.1% based on ResNet-50.

Third, as mentioned in the last section, it’s 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 $A = A_s$ ($A_s$ comes from equation 6), $A^{ij}_s$ 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 $\gamma$ 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 $\gamma$. As shown in Table 2, there is a little difference in performance with different $\gamma$. We think that the learning weights matrix $W_d$ counteract the influence of different $\gamma$. But we find that a small or a large $\gamma$ 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 obtain a 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%.

Comparison of Parameters Compared with the state-of-the-art method (Kreiss, Bertoni, and Alahi 2019) in bottom-up methods, as shown in Table
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 $F_{16}$.

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


