Graph-PCNN: Two Stage Human Pose Estimation with Graph Pose Refinement

Jian Wang*, Xiang Long*, Yuan Gao, Errui Ding, and Shilei Wen

Department of Computer Vision Technology(VIS), Baidu Inc.
{wangjian33,longxiang,gaoyuan18,dingerrui,wenshilei}@baidu.com

Abstract. Recently, most of the state-of-the-art human pose estimation methods are based on heatmap regression. The final coordinates of keypoints are obtained by decoding heatmap directly. In this paper, we aim to find a better approach to get more accurate localization results. We mainly put forward two suggestions for improvement: 1) different features and methods should be applied for rough and accurate localization, 2) relationship between keypoints should be considered. Specifically, we propose a two-stage graph-based and model-agnostic framework, called Graph-PCNN, with a localization subnet and a graph pose refinement module added onto the original heatmap regression network. In the first stage, heatmap regression network is applied to obtain a rough localization result, and a set of proposal keypoints, called guided points, are sampled. In the second stage, for each guided point, different visual feature is extracted by the localization subnet. The relationship between guided points is explored by the graph pose refinement module to get more accurate localization results. Experiments show that Graph-PCNN can be used in various backbones to boost the performance by a large margin. Without bells and whistles, our best model can achieve a new state-of-the-art 76.8% AP on COCO `test-dev` split.

Keywords: Human Pose Estimation, Keypoint Localization, Two Stage, Graph Pose Refinement

1 Introduction

Human pose estimation[1] is a fundamental yet challenging computer vision problem, that aims to localize keypoints (human body joints or parts). It is the basis of other related tasks and various downstream vision applications, including video pose estimation[43], tracking[10,42] and human action recognition [22,39,44]. This paper is interested in 2D pose estimation to detect the spatial location (i.e. 2D coordinate) of keypoints for persons in a top-down manner. Keypoint localization is a very challenging task, even for humans. It is really difficult to locate the keypoint coordinates precisely, since the variation of clothing, the occlusion between the limbs, the deformation of human joints under different poses and the complex unconstrained background, will affect the keypoint recognition and localization [49].

* Both authors contributed equally to this work.

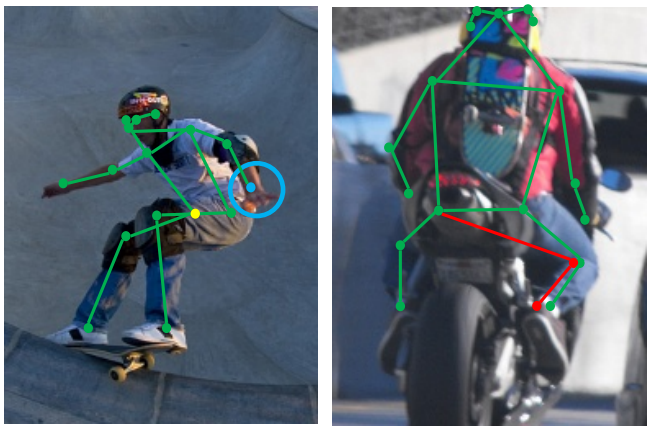

Fig. 1. Example of 2D pose estimation. The green points and lines indicate keypoints and their connections that are correctly predicted, while the red ones indicate incorrect predictions. We have observed two important characteristics of keypoint localization: 1) different features and processes are preferred for rough and accurate localization, 2) relationship between keypoints should be considered.

Most existing state-of-the-art methods use CNNs to get the heatmap of each keypoint [7,16,40,41,12,46,11,34,45,42,29,36,8,20,25,32,33]. Then the heatmap will be directly decoded to keypoint coordinates. However, these approaches do not take into account two important characteristics of human pose estimation: 1) different features and processes are preferred for rough and accurate localization, 2) relationship between keypoints should be considered.

First, humans perform keypoint localization in a two-step manner[15], [4], [2], [26]. For example, for the blue point in Fig. 1(a), we will first perform a rough localization based on the context information, including fingers and arms shown in the blue circle, to determine whether there is a wrist keypoint in a nearby area. This step can be treated as a proposal process. After rough localization, we will further observe the detail structure of the wrist itself to determine the accurate location of wrist keypoint, which can be seen as a refinement process.

We get inspiration from object detection. In object detection methods, proposal and refinement are performed based on two different feature map achieved by two separate subnets. We suggest that the proposal and refinement processes in keypoint localization should also be based on different feature maps. Therefore, we apply two different subnets to get feature maps for proposal and refinement respectively. Besides, two-stage method is very common in object detection, and can achieve excellent results in terms of both effectiveness and performance. A natural idea is that we apply the design of the two-stage to keypoint localization task, let the first stage focus on the proposal process, improving the recall of keypoint, and let the second stage focus on the refinement

process, improving the localization accuracy. Therefore, we introduce the concept of Guided Point. First, we select the guided points based on the heatmap as rough proposals in the first stage. Then in the second stage, based on the corresponding features of selected guided points, we perform coordinate refinement for accurate keypoint regression.

Secondly, in the case of complicated clothing and occlusion, the relationship between the keypoints is very important to judge its location. For example, the yellow keypoint in Fig. 1(a), due to the occlusion, we cannot see its location directly. We can only infer it from the location of other related keypoints. In addition, due to the structural limitations of the human body, there exist obvious mutual constraints between keypoints. In the refinement process, considering the relationship between keypoints may help to avoid and correct the misprediction. For example, in Fig. 1(b), the keypoints in red are the wrong predictions of the left leg. By considering the connection between them and other keypoints, we can find out and correct these errors more easily.

However, in the traditional heatmap based method, we cannot know the location of keypoints before decoding the heatmap to coordinates. This makes it difficult for us to build a pose graph that connects keypoint features at different locations. After introducing guided points, we can know the rough locations of keypoints, such that we can build a pose graph between keypoints easily. Therefore, we propose a graph pose refinement (GPR) module, which is an extension of graph convolutional network, to improve the accuracy of keypoint localization.

The main contributions of this paper include:

- This paper proposes a model-agnostic two-stage keypoint localization framework, Graph-PCNN, which can be used in any heatmap based keypoint localization method to bring significant improvement.
- A graph pose refinement module is proposed to consider the relationship between keypoints at different locations, and further improve the localization accuracy.
- Our method set a new stage-of-the-art on COCO `test-dev` split.

2 Related work

The classical approach of human pose estimation is using the pictorial structures framework with a pre-defined pose or part templates not depending on image data, which limit the expressiveness of the model [47,31].

Convolution Neural Networks (CNNs) have dramatically changed the direction of pose estimation methods. Since the introduction of "DeepPose" [38] by Toshev et al., most recent pose estimation systems have generally adopted CNNs as their backbone. There are mainly two kinds of methods to get the locations of keypoints: directly regress coordinates and estimate the heatmaps of the keypoints first, and then decode to coordinates.

Coordinate based Methods Only a few methods regress coordinates of keypoints directly. DeepPose [38] formulate pose estimation as a CNN-based regression problem directly towards body joints in a holistic fashion. Fan et al.,

[13] propose to integrate both the body part appearance and the holistic view of each local part for more accurate regression. A few other methods [6,35] further improve performance, but there is still a gap between with heatmap based methods.

Heatmap based Methods The heatmap representation is first introduce by Tompson et al. [37], and then quickly becomes the most popular solution in state-of-the-art methods. A lot of research works improve the network architectures to improve the effectiveness of heatmap regression [7,16,3,23,28,40,41,12,46,11,34,45,42,29,36,8,20,25,32,33]. For example, Hourglass [28] and its follow-ups [45,9,12] consist of blocks of several pooling and upsampling layers, which looks like an hourglass, to capture information at every scale. SimpleBaseline [42] adds several deconvolutional layers to enlarge the resolution of output feature maps, which is quite simple but performs better. The HRNet [33] model has outperformed all existing methods on public dataset by maintaining a high-resolution representation through the whole process.

Hybrid Methods Some works speculate that heatmap will introduce a statistical error and try to combine heatmap estimation with coordinate offset regression for better localization accuracy [30,18]. But in these methods, heatmap estimation and coordinate regression are performed at the same time on the same feature map, without refinement process to gradually improve accuracy.

Refinement Methods Many works focus on coordinate refinement to improve the accuracy of keypoints localization[6,4,15,26]. Instead of predicting absolute joint locations, Carreira et al. refine pose estimation by predicting error feedback at each iteration[6], Bulat et al. design a cascaded architecture for mining part relationships and spatial context[4]. Some other works use a human pose refinement network to exploit dependencies between input and output spaces [15,26]. However, they can not effectively combine heatmap estimation and coordinate regression, and the relationship between different keypoints is not considered during refinement. Our method will introduce the relationship between keypoints for more effective refinement. Zhang et al. [50] builds a pose graph directly on heatmaps and uses Graph Neural Network for refinement. However, it essentially only considers the relationship between heatmap weights at the same location, while the visual information of keypoints is completely ignored. In our framework, pose graph is built on the visual features at the position of corresponding keypoints, which is more conducive to subsequent refinement.

3 Two stage pose estimation framework

In the top-down manner pose estimation methods, single person pose estimator aims to locate K keypoints $\mathbf{P} = f\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_k g$ from an image $\mathbf{I}$ of size $W \quad H \quad 3$, where $\mathbf{p}_k$ is a 2D-coordinates. Heatmap based methods transform this problem to estimating K heatmaps $f\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_k g$ of size $W^0 \quad H^0 \quad K$, where each heatmap $\mathbf{H}_k$ will be decoded to the corresponding coordinates $\mathbf{p}_k$ during the test phase.

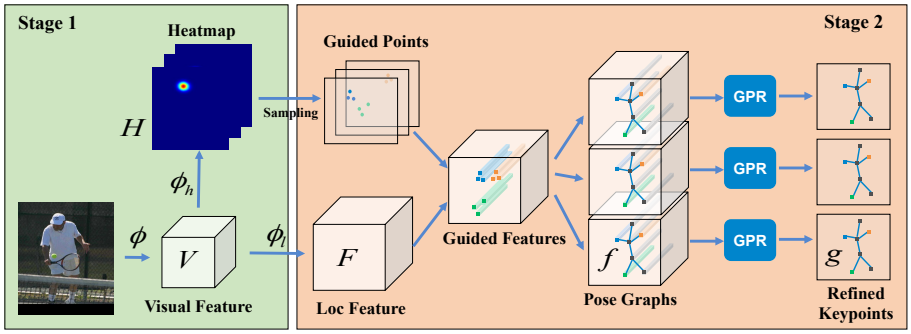

Fig. 2. Overall architecture of two stage pose estimation framework. In the first stage, heatmap regressor is applied to obtain a rough localization heatmap, and a set of guided points are sampled. In the second stage, guided points with corresponding localization features are constructed as pose graphs and then feed into a graph pose refinement (GPR) module to get refined results.

Our method simply follows the popular methods to generate the heamap in the first stage. A common pipeline is first to use a deep convolutional network ϕ to extract visual features $\mathbf{V}$ from image $\mathbf{I}$,

$$\mathbf{V} = \phi(\mathbf{I}). \tag{1}$$

A heapmap regressor ϕ_h, typically ended with a 1 1 convolutional layer, is applied to estimating the heatmaps,

$$f\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_k g = \phi_h(\mathbf{V}). \tag{2}$$

The refinement network is added after the heatmap regression, without any changes to the existing network architecture in the first stage. Therefore, our method can be applied to any heatmap based models easily. The overall architecture of our method is shown in Fig. 2. At first, we apply a localization subnet ϕ_l to transform the visual feature to the same spacial scale as heatmaps,

$$\mathbf{F} = \phi_l(\mathbf{V}), \tag{3}$$

where the size of $\mathbf{F}$ is W^0 H^0 C. During training, N guided points $f\mathbf{s}_k^1, \mathbf{s}_k^2, ..., \mathbf{s}_k^N g$ are sampled for each heatmap $\mathbf{H}_k$, while the best guided points $\mathbf{s}_k$ is selected for heapmap $\mathbf{H}_k$ during testing. For sake of simplification, we omit the superscript in the following formula. For any guided point $\mathbf{s}_k$, guided feature $\mathbf{f}_k = \mathbf{F}[\mathbf{s}_k]$ at the corresponding location and its confidence score $h_k = \mathbf{H_k}[\mathbf{s}_k]$ can be extracted.

Subsequently, we can build N pose graph for N K guided features, and introduce a graph pose refinement (GPR) module to refine the visual features by considering the relationship between keypoints.

$$f\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_K g = \mathrm{GPR}(f\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_K g, fh_1, h_2, ..., h_K g). \tag{4}$$

Fig. 3. Illustration of sampling region. Taking right wrist as example, (a), (b), (c) show the three kinds of guided points respectively, points which are close to the ground truth keypoint, points which are far away from the ground truth keypoint and points which have high heat response, where the yellow circle points indicate sampled guided points and the yellow star points indicate ground truth of right wrist.

Finally, the refined classification result $\mathbf{c}_k$ and offset regression result $\mathbf{r}_k$ are achieved based on the refined feature $\mathbf{g}_k$. The refined coordinate of keypoint is

$$\mathbf{p}_k = \mathbf{s}_k + \mathbf{r}_k. \tag{5}$$

In the following, we first describe the guided point sampling strategy in section 3.1. Second, we show the detail structure of graph pose refinement module in section 3.2. Third, we introduce the loss used for training in section 3.3. Finally, we show how to integrate our framework to existing backbones and elaborate the details of training and testing in section 3.4.

3.1 Guided Point Sampling

Locating human joint based on the peak of heatmap is frequently-used in modern human pose estimators, and they modeled the target of heatmap by generating gaussian distribution around the ground truth. But due to the complex image context and human action, the joint heat may not be satisfy gaussian distribution strictly which, together with quantisation affect of image resolution downsampling, leads to an insufficient precision of this localization method. However, the peak of heatmap is always close to the true location of joint, which make it adequate to regress the true location.

To achieve the goal of obtaining refined coordinate based on the peak of heatmap, we sample several guided points and train coordinate refinement in stage2. Concretely, we equally sample three kinds of guided points for training: (a) points which are close to the ground truth keypoint, (b) points which are far away from the ground truth keypoint, and (c) points which have high heat response. And the kth ground truth keypoint is denoted as $\mathbf{t}_k$. As exhibited in

Fig. 3, (a) and (b) are randomly sampled within the red region and blue region, respectively, and the red region which centered at ground truth has a radius of 3σ, where σ is same with the standard deviation for generating gaussian heatmap target. (c) is randomly sampled from the top N highest response points at the heatmap.

Due to the different characterization of different keypoints, we sample guided points for each keypoint individually, and the total amount of the three kinds of guided points for each keypoint is set equally to N.

After the N guided points $f\mathbf{s}_k^1, \mathbf{s}_k^2, ..., \mathbf{s}_k^N g$ are sampled, we divide them into two sets, positive set and negative set, denoted as

$$\begin{aligned} S_k^+ &= f\mathbf{s}_k \; j \; jj\mathbf{s}_k \quad \mathbf{t}_k jj < 3\sigma g \\ S_k &= f\mathbf{s}_k \; j \; jj\mathbf{s}_k \quad \mathbf{t}_k jj \quad 3\sigma g \end{aligned} \tag{6}$$

and $N_k^+ = jS_k^+ j$, $N_k = jS_k j$. Then all of the corresponding guided feature extracted from $\mathbf{F}$ by means of bilinear interpolation are feeded into stage2 for refinement while only the guided points from positive set contributed to the coordinate regression.

According to the above label assignment manner, (a) and (b) are definite positive and negative samples, and the influence of proportion between them will be explored in Section 4.3. While (c) is almost negative samples during the beginning stage of training and turns to positive samples as the training schedule goes on. We suppose that (c) can not only accelerate the feature learning at the beginning of training, because (c) are hard negative samples for classification at this stage, but also contribute to the learning of regression when the classification status of feature is relatively stable, as (c) are almost positive samples at this period. Further more, (c) is not necessarily positive when the model converges roughly because of some prediction error caused by hard situation. In this circumstances, (c) can also be regarded as hard negative samples for helping the model to be trained better.

3.2 Graph Pose Refinement

In most of previous works, many fields have been well studied for human pose estimation, such as network structure, data preprocessing and postprocessing, post refinement, etc. However, in these works, the localization of human keypoints are conducted independently for each keypoint while the relationship between different keypoints is ignored all along. Intuitively, the human keypoints construct a salient graph structure base on the pattern of human body, and they have clear adjacent relation with each other. So we consider that the localization of keypoints can be infered better with the help of the information hinted by this relationship. For instance, in our framework, if we know that a guided point is left elbow, then the positive guided points of left wrist should tend to have higher response on left wrist a priori, as left wrist is adjacent to left elbow. So

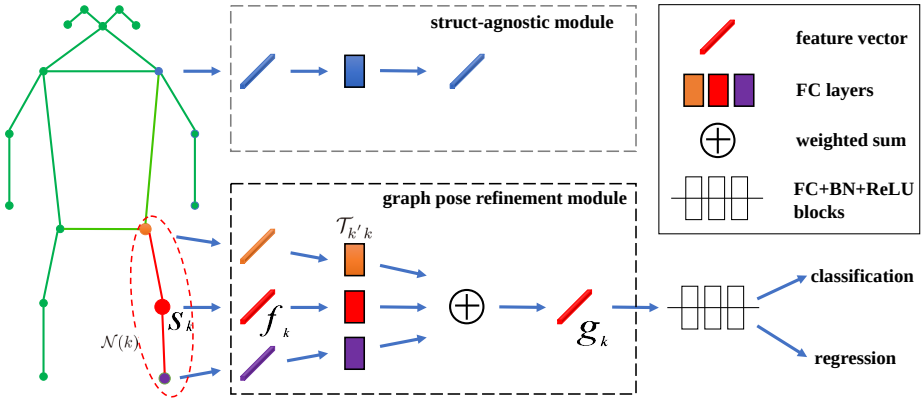

Fig. 4. The structure of graph pose refinement module. The relationship between keypoints is taken into account in contrast to the struct-agnostic module.

that more supervision can be imposed upon the feature of these keypoints than treating them independently.

To take advantage of the information implicit in the graph structure mentioned above, we propose a graph pose refinement module to model it, and then refine the feature of these keypoints. As shown in Fig. 4, we build a graph and conduct graph convolution for each keypoint. The output embedding feature can be computed by

$$\begin{aligned} \mathbf{g}_k &= \frac{1}{Z_k} \sum_{\mathbf{s}_{k'} \in \mathcal{N}(k)} \omega_{k'} \mathcal{T}_{k'k}(\mathbf{f}_{k'}) \\ \omega_{k'} &= \begin{cases} h_{k'} \mathbb{1}(R_{k'}), & k' \neq k \\ 1, & k' = k \end{cases} \end{aligned} \tag{7}$$

where $\mathcal{N}(k)$ represents for a point set containing the guided point $\mathbf{s}_k$ and its neighbours, $\mathcal{T}_{k'k}$ for the linear transformation from guided point $\mathbf{s}_{k'}$ to $\mathbf{s}_k$, and $\mathbb{1}$ for the indicator function. $Z_k = \sum_{\mathbf{s}_{k'} \in \mathcal{N}(k)} \omega_{k'}$ is used for normalization. $R_{k'}$ is a boolean type parameter encoding the reliability of a guided point which works for filtering out points of low quality, and its definition will be explained in detail in section 3.4.

Specially, as defined in (7), this graph convolution is an extension of traditional graph convolution, it is designed by considering the characteristic of pose estimation problem. Firstly, we add a weight for each message generated from $\mathbf{s}_{k'}$ to $\mathbf{s}_k$, which can control the contribution of each message according to the intensity and reliability of $\mathbf{s}_{k'}$. With the constraint of these weights, the graph convolution can be trained more stably. Further more, we set $\omega_{k'} = 1$ when $k' = k$. And this can make the graph convolution degrading to a traditional

linear transformation for $\mathbf{s}_k$ when $\mathbb{1}(R_{k'}) = 0$ for all $\mathbf{s}_{k'}$ 2 $N(k)$ where k' 6= k, without being affected by the intensity and reliability of $\mathbf{s}_k$ itself.

3.3 Loss Function

After the refinement module above, the embedded feature is sent to a module containing several fully connected layers and batch norm layers, as illustrated in Fig. 4. Finally two predictions are outputed, denoted as $\mathbf{c}_k$ and $\mathbf{r}_k$, for classification and regression respectively. Giving ground truth keypoint location $\mathbf{t}_k$, the losses for these two branches are defined as

$$L_k^{cls} = \frac{1}{2}\left[\frac{1}{N_k^+}\sum_{\mathbf{s}_k^i 2 S_k^+}\alpha_k^i L_{cls}(\mathbf{c}_k^i, 1) + \frac{1}{N_k}\sum_{\mathbf{s}_k^i 2 S_k} L_{cls}(\mathbf{c}_k^i, 0)\right] \tag{8}$$

$$\alpha_k^i = exp(\quad \frac{(\mathbf{s}_k^i \quad \mathbf{t}_k)^2}{2\sigma^2})$$

and

$$L_k^{reg} = \frac{1}{N_k^+}\sum_{\mathbf{s}_k^i 2 S_k^+} L_{reg}(\mathbf{r}_k^i, \mathbf{t}_k \quad \mathbf{s}_k^i), \tag{9}$$

where L^{cls} and L^{reg} are softmax cross-entropy loss and L1 loss. The total loss of the stage2, can be expressed as

$$L^{s_2} = \frac{\sum_k \gamma_k (L_k^{cls} + \lambda L_k^{reg})}{\sum_k \gamma_k}, \tag{10}$$

where γ_k is the target weight of keypoint k. And λ is a loss weight which is set to 16 constantly. And the total loss of Graph-PCNN is

$$L = L^{s_1} + L^{s_2}, \tag{11}$$

where L^{s_1} is the traditional heatmap regression loss for stage1.

3.4 Network Architecture

Network Architecture In previous works such as [30], [18], there is also a coordinate refinement after heatmap decoding, and their coordinate refinement branch share the same feature map with heatmap prediction branch. However, the rough and accurate localization always need different embedding feature, further more, it is hard to conduct particular feature refinement for either of these two branches. In order to alleviate the above problems, we copy the last stage of the backbone network to produce two different feature maps with the same size followed by heatmap regression convolution and graph pose refinement module respectively. By means of this modification, the network can learn more particular feature for two different branches, and easily conduct guided points sampling for further feature refinement.

Training and Testing For the proposed two stage pose estimation framework, several operations are specific in the training and testing phase.

Firstly, in order to make the stage2 be trained sufficiently, we sample multiple guided points for each keypoint following the strategy described in Section 3.1 during training, and the amount of guided points N is various according to the input size. While during testing, only one guided point is generated by decoding the predicted heatmap, and the output score of it is gathered as the corresponding heat response score from stage1. Following most of previous works[42], [33], a quarter offset in the direction from the highest response to the second highest response is added to the position of heatmap peak for higher precision, when decoding this guided point from heatmap.

Secondly, the definition of guided point reliability metric R_{k^0} is different for training and testing, which is represented as

$$R_{k^0} = \begin{cases} jj\mathbf{s}_{k^0} \quad \mathbf{t}_{k^0}jj < \delta & in\ training\ phase \\ h_{k^0} > \xi & in\ testing\ phase \end{cases} \tag{12}$$

At the training phase, the ground truth is available for measuring this reliability, and the guided points which are close to their corresponding ground truth can be regarded reliable. δ is a distance threshold controling the close degree which equals to 2σ. While at the testing phase the ground truth is unknown, so for insurance, the guided points which heat responses are high enough are qualified to pass message to their neighbour points. And ξ is a threshold for gating the heat response, which is set to 0.85 constantly.

Finally, during training, we shuffle the guided points of one keypoint after the guided point sampling in order to create more various situation of graph combination, which can make the graph pose refinement module more generalized.

4 Experiments

4.1 Dataset

In this paper, we use the most popular human pose estimation dataset, COCO. The COCO keypoint dataset [24] presents challenging images with multi-person pose of various body scales and occlusion patterns in unconstrained environments. It contains 200,000 images and 250,000 person samples. Each person instance is labelled with 17 joints. We train our models on `train2017`(includes 57K images and 150K person instances) with no extra data, and conduct ablation study on `val2017`. Then we test our models on `test-dev` for comparison with the state-of-the-art methods. In evaluation, we use the metric of Object Keypoint Similarity (OKS) for COCO to report the model performance.

4.2 Implementation Details

For fair comparison, we follow the same training configuration as [42] and [33] for ResNet and HRNet respectively. To construct the localization subnet, we

Table 1. Ablation study on COCO `val2017`

Method	Size	stage1 AP	stage2 AP
SBN	128x96	59.3	-
Graph-PCNN	128x96	**61.1**	**64.6**
SBN	256x192	70.4	-
Graph-PCNN	256x192	**71.3**	**72.6**
SBN	384x288	72.2	-
Graph-PCNN	384x288	**72.7**	**73.6**

copy the conv5 stage, which spatial size is 1/32 to the input size, and the last three deconvolution layers for ResNet series networks, while copying the stage4, which has three high resolution modules, for HRNet series networks. For ablation study, we also add 128x96 input size in our experiment following [49]. And we set N as 48, 192 and 432 corresponding to the three input sizes of 128x96, 256x192 and 384x288 during all of our experiment except the ablation study of N. During inference, we use person detectors of AP 56.4 and 60.9 for COCO `val2017` and `test-dev` respectively, while for pose estimation, we evaluate single model and only use flipping test strategy for testing argumentation.

4.3 Ablation Studies

We use ResNet-50 backbone to perform ablation study on COCO `val2017`.

Two stage pose estimation framework. Firstly, we evaluate the effectiveness of our proposed two stage pose estimation framework. As Table 1 shows, the stage2 of Graph-PCNN gives 5.3%, 2.2%, 1.4% AP gain comparing to original simple baseline network(SBN) at the three input sizes, which demonstrates that our regression based two stage framework is more effective than decoding joint location from heatmap. Further more, we test the stage1 of Graph-PCNN which shares the same network architecture with SBN. It should be noted that training with Graph-PCNN can also boost the performance of heatmap, and 1.8%, 0.9%, 0.5% AP gain are got as shown. That means we can also get considerable performance boosting without any extra computing cost during inference if we only use the stage1 of Graph-PCNN.

Sampling strategy. Secondly, we study the influence of the proportion of different kinds of guided points and the total amount of guided points N based on ResNet-50 with 128x96 input size. In order to avoid exploring the proportion among all the three kinds of guided points, we simplify the proportion study by using only definite positive points and negative points, and then we set different proportion between them with N unchanged. From the results shown in Fig. 5 (a), we can come to that the proportion ranging from 1:2 to 2:1 is already appropriate, and the sampling strategy proposed in Section 3.1 can fit this proportion range in any situation. In addition, we try different N based on the strategy in Section 3.1 and finally select 48 as the value of N according to the results reported in Fig. 5 (b).

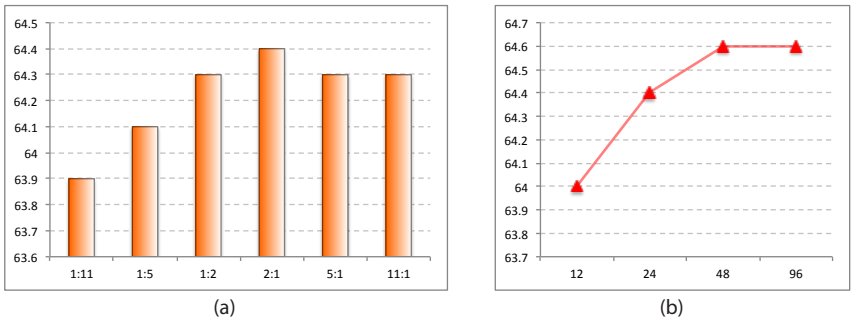

Fig. 5. Influence of the proportion and total amount of guided points in sampling. (a) is the results on different proportions, while the x-axis represents the proportions between positive guided points and negative guided points. (b) is the results on different values of the total amount, while the x-axis represents the values of N.

Table 2. Effectiveness of the graph pose refinement(GPR) module.

Method	Size	stage1 AP	stage2 AP
struct-agnostic	128x96	60.7	63.8
GPR-va	128x96	61.2	52.1
GPR-vb	128x96	61.1	64.5
GPR-vc	128x96	60.8	64.3
GPR	128x96	61.1	**64.6**

Graph pose refinement module. Finally, we evaluate the contribution of the proposed graph pose refinement(GPR) module. In this study, we compare proposed GPR with a struct-agnostic baseline module and several variants of GPR(GPR-va, GPR-vb, GPR-vc). GPR-va set $\omega_{k^0} = 1$ for all $fk^0 js_{k^0}\ 2\ N(k)g$ in (7), GPR-vb set $\omega_{k^0} = \mathbb{1}(R_{k^0})$ for $fk^0 js_{k^0}\ 2\ N(k), k^0\ 6=\ kg$ with the heat response factor dropped, and GPR-vc dropped the guided points shuffling operation mentioned in Section 3.4. The comparison results are displayed in Table 2. We can see that GPR boosts the stage1 AP and stage2 AP by 0.4% and 0.8% respectively, comparing to the struct-agnostic baseline. And the performance of GPR is better than all of its other variants, which reveals the importance of parameter ω_{k^0} and the guided points shuffling operation. Especially, the reliability factor $\mathbb{1}(R_{k^0})$ affects the performance greatly. Thus, we believe that GPR can refine the feature of a guided point by taking advantage of the supervision signal of its neighbour keypoint which is good located, as we supposed in Section 3.2.

4.4 Comparison with Other Methods with Coordinate Refinement

DARK[49] is a state-of-the-art method which improved traditional decoding by a more precise refinement based on Taylor-expansion. We follow the training settings of DARK and compare our refinement results with it. From Table 3

Table 3. Comparison with distribution-aware coordinate representation of keypoint(DARK) on COCO `val2017`.

Method	Backbone	Size	*AP*	AP^{50}	AP^{75}	AP^{M}	AP^{L}	*AR*
DARK	R50	128x96	62.6	86.1	70.4	60.4	67.9	69.5
Graph-PCNN	R50	128x96	**64.6**	**86.4**	**72.7**	**62.4**	**70.1**	**71.5**
DARK	R101	128x96	63.2	86.2	71.1	61.2	68.5	70.0
Graph-PCNN	R101	128x96	**64.8**	**86.6**	**73.1**	**62.6**	**70.3**	**71.7**
DARK	R152	128x96	63.1	86.2	71.6	61.3	68.1	70.0
Graph-PCNN	R152	128x96	**66.1**	**87.2**	**74.6**	**64.1**	**71.5**	**73.0**
DARK	HR32	128x96	70.7	88.9	78.4	67.9	76.6	76.7
Graph-PCNN	HR32	128x96	**71.5**	**89.0**	**79.0**	**68.4**	**77.6**	**77.3**
DARK	HR48	128x96	71.9	89.1	79.6	69.2	78.0	77.9
Graph-PCNN	HR48	128x96	**72.8**	**89.2**	**80.1**	**69.9**	**79.0**	**78.6**
DARK	HR32	256x192	75.6	90.5	82.1	71.8	82.8	80.8
Graph-PCNN	HR32	256x192	**76.2**	**90.3**	**82.6**	**72.5**	**83.2**	**81.2**
DARK	HR32	384x288	76.6	90.7	82.8	72.7	83.9	81.5
Graph-PCNN	HR32	384x288	**77.2**	**90.7**	**83.6**	**73.5**	**84.0**	**82.1**

Table 4. Comparison with model-agnostic human pose refinement network(PoseFix) on COCO `val2017`.

Method	Backbone	Size	*AP*	AP^{50}	AP^{75}	AP^{M}	AP^{L}	*AR*
PoseFix	R50	256x192	72.1	88.5	78.3	68.6	78.2	-
Graph-PCNN	R50	256x192	**72.6**	**89.1**	**79.3**	**69.1**	**79.7**	78.1

we can observe that our Graph-PCNN generally outperforms DARK over different network architecture and input size. This suggests that regression based refinement predicts coordinate more precise than analyzing the distribution of response signal from heatmap, as the response signal itself may not satisfy gaussian distribution strictly because of complex human pose and image context while regression is regardless of these drawback.

PoseFix[26] is a model-agnostic method which refines a existing pose result from any other method by a independent model. A coarse-to-fine coordinate estimation schedule ended by coordinate calculation following integral loss[35] is used to enhance the precision. We conduct comparison with PoseFix by using same backbone and input size with its model from refinement stage and the performance of human detectors for these two methods are comparable, AP 55.3 vs 56.4 for PoseFix(using CPN) and our Graph-PCNN respectively. As illustraed in Table 4, we achieve a competable result with PoseFix, but PoseFix included input from CPN which need an extra R50 network while our method only need an extra R50 conv5 stage as refinement branch.

Table 5. Comparison with the state-of-the-arts methods on COCO `test-dev`.

Method	Backbone	Size	*AP*	AP^{50}	AP^{75}	AP^{M}	AP^{L}	*AR*
CMU-Pose[5]	-	-	61.8	84.9	67.5	57.1	68.2	66.5
Mask-RCNN[17]	R50-FPN	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI[30]	R101	353x257	64.9	85.5	71.3	62.3	70.0	69.7
AE[27]	-	512x512	65.5	86.8	72.3	60.6	72.6	70.2
Integral Pose[35]	R101	256x256	67.8	88.2	74.8	63.9	74.0	-
CPN[8]	ResNet-Inception	384x288	72.1	91.4	80.0	68.7	77.2	78.5
RMPE[14]	PyraNet[45]	320x256	72.3	89.2	79.1	68.0	78.6	-
CFN[19]	-	-	72.6	86.1	69.7	78.3	64.1	-
CPN(ensemble)[8]	ResNet-Inception	384x288	73.0	91.7	80.9	69.5	78.1	79.0
Posefix[26]	R152+R152	384x288	73.6	90.8	81.0	70.3	79.8	79.0
CSM+SCARB[32]	R152	384x288	74.3	91.8	81.9	70.7	80.2	80.5
CSANet[48]	R152	384x288	74.5	91.7	82.1	71.2	80.2	80.7
MSPN[21]	MSPN	384x288	76.1	93.4	83.8	72.3	81.5	81.6
Simple Base[42]	R152	384x288	73.7	91.9	81.1	70.3	80.0	79.0
UDP[18]	R152	384x288	74.7	91.8	82.1	71.5	80.8	80.0
Graph-PCNN	R152	384x288	**75.1**	**91.8**	**82.3**	**71.6**	**81.4**	**80.2**
HRNet[33]	HR32	384x288	74.9	92.5	82.8	71.3	80.9	80.1
UDP[18]	HR32	384x288	76.1	92.5	83.5	72.8	82.0	81.3
Graph-PCNN	HR32	384x288	**76.4**	**92.5**	**83.8**	**72.9**	**82.4**	**81.3**
HRNet[33]	HR48	384x288	75.5	92.5	83.3	71.9	81.5	80.5
DARK[49]	HR48	384x288	76.2	92.5	83.6	72.5	82.4	81.1
UDP[18]	HR48	384x288	76.5	**92.7**	84.0	73.0	82.4	81.6
PoseFix[26]	HR48+R152	384x288	76.7	92.6	84.1	73.1	82.6	81.5
Graph-PCNN	HR48	384x288	**76.8**	92.6	**84.3**	**73.3**	**82.7**	**81.6**

4.5 Comparison to State of the Art

We compare our Graph-PCNN with other top-performed methods on COCO `test-dev`. As Table 5 reports, our method with HR48 backbone at the input size of 384x288 achieves the best AP(76.8), and improves HR48 with the same input size(75.5) by a large margin(+1.3). Mean while, It also outperforms other competitors with same backbone and input size settings, such as DARK(76.2), UDP(76.5) and PoseFix(76.7), which illustrates the advantages of our method.

5 Conclusions

In this paper, we propose a two stage human pose estimator for the top-down pose estimation network, which improves the overall localization performance by introducing different features for rough and accurate localization. Meanwhile, a graph pose refinement module is proposed to refine the feature for pose regression by taking the relationship between keypoints into account, which boosts the performance of our two stage pose estimator further. Our proposed method is model-agnostic and can be added on most of the mainstream backbone. Even better, more improvement can be explored by drawing on the successful experience of the two stage detection framework in the future.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: CVPR (2014)
2. Belagiannis, V., Rupprecht, C., Carneiro, G., Navab, N.: Robust optimization for deep regression. In: Proceedings of the IEEE international conference on computer vision. pp. 2830–2838 (2015)
3. Belagiannis, V., Zisserman, A.: Recurrent human pose estimation. In: FG (2017)
4. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: European Conference on Computer Vision. pp. 717–732. Springer (2016)
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)
6. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. In: CVPR (2016)
7. Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NeurIPS (2014)
8. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: CVPR (2018)
9. Chen, Y., Shen, C., Wei, X.S., Liu, L., Yang, J.: Adversarial posenet: A structure-aware convolutional network for human pose estimation. In: ICCV (2017)
10. Cho, N.G., Yuille, A.L., Lee, S.W.: Adaptive occlusion state estimation for human pose tracking under self-occlusions. Pattern Recognition **46**(3), 649–661 (2013)
11. Chu, X., Ouyang, W., Li, H., Wang, X.: Structured feature learning for pose estimation. In: CVPR (2016)
12. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X.: Multi-context attention for human pose estimation. In: CVPR (2017)
13. Fan, X., Zheng, K., Lin, Y., Wang, S.: Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In: CVPR (2015)
14. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: ICCV (2017)
15. Fieraru, M., Khoreva, A., Pishchulin, L., Schiele, B.: Learning to refine human pose estimation. In: CVPR (2018)
16. Gkioxari, G., Toshev, A., Jaitly, N.: Chained predictions using convolutional neural networks. In: ECCV (2016)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
18. Huang, J., Zhu, Z., Guo, F., Huang, G.: The devil is in the details: Delving into unbiased data processing for human pose estimation. arXiv preprint arXiv:1911.07524 (2019)
19. Huang, S., Gong, M., Tao, D.: A coarse-fine network for keypoint localization. In: ICCV (2017)
20. Ke, L., Chang, M.C., Qi, H., Lyu, S.: Multi-scale for human pose estimation. In: ECCV (2018)
21. Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, G., Lu, H., Wei, Y., Sun, J.: Rethinking on multi-stage networks for human pose estimation. arXiv preprint arXiv:1901.00148 (2019)
22. Liang, Z., Wang, X., Huang, R., Lin, L.: An expressive deep model for human action parsing from a single image. In: ICME. IEEE (2014)

23. Lifshitz, I., Fetaya, E., Ullman, S.: Human pose estimation using deep consensus voting. In: ECCV (2016)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
25. Liu, W., Chen, J., Li, C., Qian, C., Chu, X., Hu, X.: A cascaded inception of inception network with attention modulated feature fusion for human pose estimation. In: AAAI (2018)
26. Moon, G., Chang, J.Y., Lee, K.M.: Posefix: Model-agnostic general human pose refinement network. In: CVPR (2019)
27. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: NeurIPS (2017)
28. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV (2016)
29. Ning, G., Zhang, Z., He, Z.: Knowledge-guided deep fractal neural networks for human pose estimation. IEEE Transactions on Multimedia **PP**(99), 1–1 (2017)
30. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K.: Towards accurate multi-person pose estimation in the wild. In: CVPR (2017)
31. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: CVPR (2013)
32. Su, K., Yu, D., Xu, Z., Geng, X., Wang, C.: Multi-person pose estimation with enhanced channel-wise and spatial information. In: CVPR (2019)
33. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019)
34. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: ICCV (2017)
35. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: ECCV (September 2018)
36. Tang, W., Yu, P., Wu, Y.: Deeply learned compositional models for human pose estimation. In: ECCV (2018)
37. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: NeurIPS (2014)
38. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: CVPR (2014)
39. Wang, C., Wang, Y., Yuille, A.L.: An approach to pose-based action recognition. In: CVPR (2013)
40. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016)
41. Xiao, Ouyang, W., Wang, X., et al.: Crf-cnn: Modeling structured information in human pose estimation. In: NeurIPS (2016)
42. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: ECCV (2018)
43. Xiaohan Nie, B., Xiong, C., Zhu, S.C.: Joint action recognition and pose estimation from video. In: CVPR (2015)
44. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI (2018)
45. Yang, W., Li, S., Ouyang, W., Li, H., Wang, X.: Learning feature pyramids for human pose estimation. In: ICCV (2017)
46. Yang, W., Ouyang, W., Li, H., Wang, X.: End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In: CVPR (2016)

47. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE transactions on pattern analysis and machine intelligence **35**(12), 2878–2890 (2012)
48. Yu, D., Su, K., Geng, X., Wang, C.: A context-and-spatial aware network for multi-person pose estimation. arXiv preprint arXiv:1905.05355 (2019)
49. Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C.: Distribution-aware coordinate representation for human pose estimation. arXiv preprint arXiv:1910.06278 (2019)
50. Zhang, H., Ouyang, H., Liu, S., Qi, X., Shen, X., Yang, R., Jia, J.: Human pose estimation with spatial contextual information. arXiv preprint arXiv:1901.01760 (2019)