InsPose: Instance-Aware Networks for Single-Stage Multi-Person Pose Estimation

Dahu Shi
Hikvision Research Institute
Hangzhou, China
shidahu@hikvision.com

Xing Wei
School of Software Engineering
Xi’an Jiaotong University
weixing@mail.xjtu.edu.cn

Xiaodong Yu
Hikvision Research Institute
Hangzhou, China
yuxiaodong7@hikvision.com

Wenming Tan
Hikvision Research Institute
Hangzhou, China
tanwenming@hikvision.com

Ye Ren
Hikvision Research Institute
Hangzhou, China
renye@hikvision.com

Shiliang Pu*
Hikvision Research Institute
Hangzhou, China
pushiliang.hri@hikvision.com

ABSTRACT
Multi-person pose estimation is an attractive and challenging task. Existing methods are mostly based on two-stage frameworks, which include top-down and bottom-up methods. Two-stage methods either suffer from high computational redundancy for additional person detectors or they need to group keypoints heuristically after predicting all the instance-agnostic keypoints. The single-stage paradigm aims to simplify the multi-person pose estimation pipeline and receives a lot of attention. However, recent single-stage methods have the limitation of low performance due to the difficulty of regressing various full-body poses from a single feature vector. Different from previous solutions that involve complex heuristic designs, we present a simple yet effective solution by employing instance-aware dynamic networks. Specifically, we propose an instance-aware module to adaptively adjust (part of) the network parameters for each instance. Our solution can significantly increase the capacity and adaptive-ability of the network for recognizing various poses, while maintaining a compact end-to-end trainable pipeline. Extensive experiments on the MS-COCO dataset demonstrate that our method achieves significant improvement over existing single-stage methods, and makes a better balance of accuracy and efficiency compared to the state-of-the-art two-stage approaches. The code and models are available at https://github.com/hikvision-research/opera.

CCS CONCEPTS
• Computing methodologies → Computer vision; Machine learning; Computer graphics.

KEYWORDS
Pose Estimation, Conditional Convolutions, Neural Networks

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Dahu Shi, Xing Wei, Xiaodong Yu, Wenming Tan, Ye Ren, and Shiliang Pu. 2021. InsPose: Instance-Aware Networks for Single-Stage Multi-Person Pose Estimation. In Proceedings of the 29th ACM Int’l Conference on Multimedia (MM ’21), Oct. 20–24, 2021, Chengdu, Sichuan Province, China. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475447

Figure 1: InsPose makes use of instance-aware keypoint networks (KP-Nets) to directly predict the full-body pose for each instance.

1 INTRODUCTION
Multi-person pose estimation aims to identify all the instances and detect the keypoints of each person simultaneously. It is a fundamental task and has a wide range of applications such as motion recognition [36, 38], person re-identification [20], pedestrian tracking [44], and athletic training assistance [37].
Multi-person pose estimation methods are usually based on two-stage frameworks including the bottom-up and top-down approaches. Bottom-up methods [3, 14, 25, 27, 28] first detect all the potential keypoints in the image in an instance-agnostic manner, and then perform a grouping process to get the final instance-aware poses. Usually, the grouping process is heuristic in which many tricks are involved and does not facilitate end-to-end learning. On the other hand, top-down methods [1, 5, 7, 10, 11, 17, 32, 40] first detect each individual by a bounding-box and then transfer the task to an easier single-person pose estimation problem. While avoiding the heuristic grouping process, top-down methods come with the price of high computational complexity as mentioned in [26]. Especially, the running time of top-down methods depends on the number of instances contained in the image. Similarly, top-down methods are also not learned in an end-to-end fashion.

Recently, the single-stage pose estimation paradigm [26, 33, 39] is proposed and receives a lot of attention. This single-stage framework receives great interest since 1) it directly learns a mapping from the input image to the desired multi-person poses and is end-to-end trainable, and 2) it can consolidate object detection and keypoint localization in a simple and unified framework, and thus 3) presents a more compact pipeline and attractive simplicity and efficiency over two-stage methods.

A straightforward way for single-stage pose estimation is adapting the bounding-box regression head of the single-stage object detectors [35, 42]. However, such a naive approach expects a single feature vector (namely, a point feature) to regress the precise locations of all the keypoints. While a point feature may be sufficient for bounding box regression to some extent, it is difficult to encode rich pose information considering that keypoints have much more geometric variations than the regular box corners. As a result, the pose estimation accuracy of this approach is unsatisfactory [33]. To deal with this problem, SPM [26] proposes a progressive joint representation by dividing body joints into hierarchies induced from articulated kinematics. CenterNet [43] proposes a hybrid solution by first regressing joint locations and then refining with the help of additional heatmaps. DirectPose [33] proposes a keypoint alignment module with the aim of aligning the point feature to the features around target joints. However, they all involve heuristic designs and still result in less accurate performance compared to the state-of-the-art bottom-up method [6], leaving room for further optimization.

In this paper, we present a direct and simple framework for single-stage multi-person pose estimation, termed InstPose. The core of our method is an instance-aware module to adaptively adjust (part of) the network parameters for each instance, as illustrated in Figure 1. Instead of using a standard keypoint network (called, KP-Nets) with fixed parameters for predicting all instances of various poses, we borrow the idea from recent dynamic neural networks [4, 9, 16, 41] to generate instance-aware parameters. Specifically, these KP-Nets are dynamically generated with respect to the different spatial locations of the input. Afterwards, they are applied to the whole feature maps to directly predict the full-body poses, get rid of the need for additional handcrafted processing. KP-Nets can better encode all kinds of pose variants since each one of them only focuses on an individual instance. Moreover, each KP-Net only consists of three $1 \times 1$ convolutions, leading to a compact and efficient architecture.

The overview of our method is illustrated in Figure 3 and we summarize our main contributions as below.

- We propose a simple single-stage network for multi-person pose estimation. It is end-to-end trainable and avoids many heuristic designs such as keypoints grouping [6], progressive pose representation [26], delicate feature aggregation [33], or pose anchors [39].
- The proposed model has a dynamic network that consists of instance-aware KP-Nets, which is different from previous methods that use a fixed set of convolutional filters for all the instances.
- We conduct extensive experiments on the COCO benchmark where our method outperforms state-of-the-art single-stage methods significantly and achieves a better trade-off between accuracy and efficiency compared to state-of-the-art two-stage approaches.

2 RELATED WORK

2.1 Multi-Person Pose Estimation

With the development of deep neural networks, modern pose estimation methods can be divided into three categories: top-down methods, bottom-up methods, and recent single-stage methods.

**Top-down Methods.** Top-down methods decouple the multi-person pose estimation task into two sub-tasks: multi-person detection and single-person pose estimation. The person detection is first performed to predict a bounding-box for each instance in the image. Then the instance is cropped from the bounding box for single-person pose estimation. Instead of cropping the original image, Mask RCNN [11] proposes to utilize extracted features to improve efficiency. A cascade pyramid network is proposed in CPN [5], aiming to regress difficult keypoints. Moreover, HRNet [32] focuses on learning reliable high-resolution representations and maintains high-resolution representations through the network. In general, top-down methods have better performance but also have higher computational complexity since they have to repeatedly perform single-person pose estimation for each instance. Moreover, top-down methods are highly dependent on the performance of the detectors.

**Bottom-up Methods.** On the other hand, bottom-up methods [3], [14], [25], [6] first detect all keypoints in instance-agnostic fashion. Then keypoints are assembled into full-body poses by a grouping process. OpenPose [3] proposes to utilize part affinity fields to establish connections between keypoints of the same instance. Associative embedding [25] produces a detection heatmap and a tagging heatmap for each body joint, and then groups body joints with similar tags into individual people. Based on the above grouping strategy, HigherHRNet [6] proposes multi-resolution heatmap aggregation for more accurate human pose estimation. Bottom-up methods are usually more efficient because of their simpler pipeline of sharing convolutional computation. However, the grouping post-process is heuristic and involves many tricks which often makes its performance inferior to top-down methods.

**Single-Stage Methods.** Both top-down and bottom-up methods are not end-to-end trainable and have their own limitations. The
single-stage methods [26], [43], [33], [39] are proposed recently to avoid the aforementioned difficulties. These methods predict instance-aware keypoints directly from estimated root locations [26], [43] or follow the dense predication strategy [39], [33] for estimating a full-body pose from each spatial location. SPM [26] proposes a structured pose representation to unify position information of person instances and body joints. DirectPose [33] and Point-set anchors [39] adopt deformable-like convolutions to refine the initial estimations, mitigating the difficulties of feature misalignment. The performance of DirectPose [33] is poor even with a well-designed feature aggregation mechanism. Although SPM and Point-set anchors achieve competitive performance among single-stage methods, they still fall behind the state-of-the-art bottom-up methods.

2.2 Dynamic Neural Networks

The dynamic neural network is an emerging research topic in deep learning. Unlike traditional neural networks which have a static model at the inference stage, dynamic networks can adapt their structures or parameters to different inputs, leading to advantages in terms of adaptiveness, capacity, efficiency, etc. A typical approach to parameter adaptation is adjusting the weights based on their input during inference. For example, conditionally parameterized convolution [41] and dynamic convolutional neural network [4] perform soft attention on multiple convolutional kernels to produce an adaptive ensemble of parameters without noticeably increasing the computational cost. On the other hand, weight prediction [9] directly generates a subset of instance-wise parameters with an independent model at test time. For instance, dynamic filter networks [16] build a generation network to predict weights for a convolutional layer. Recently, CondInst [34] employs the dynamic filters to generate binary masks for each instance and achieves promising performance in the instance segmentation field.

In this work, we leverage this idea to produce instance-aware dynamic convolutional weights to solve the challenging multi-person pose estimation problem, as shown in Figure 2. In parallel to our work, Mao et al. [24] proposed a fully convolutional multi-person pose estimator. Our method and [24] both present a single-stage multi-person pose estimation framework using dynamic instance-aware convolutions, which achieve better accuracy/efficiency trade-off than other state-of-the-art methods.

3 APPROACH

In this section, we first introduce the overall architecture of our framework. Next, we elaborate on the proposed instance-aware dynamic network, which is capable of implementing the multi-person pose estimation task. Then we describe the disk offset prediction branch to reduce the discretization error. Finally, the loss function and the inference pipeline of our model are summarized.

3.1 Overview

Let \( I \in \mathbb{R}^{H \times W \times 3} \) be an input image of height \( H \) and width \( W \), multi-person pose estimation (a.k.a. keypoint detection) aims at estimating human poses \( \mathcal{P} \) of all the person instances in \( I \). The ground-truths are defined as

\[
\mathcal{\bar{P}} = \{ \overline{P}_1, \overline{P}_2, \ldots, \overline{P}_N \}
\]

where \( N \) is the number of persons in the image, \( K \) is the number of keypoints (i.e., left shoulder, right elbow), and \( \overline{P}_i = (x_i^l, y_i^l) \) denotes coordinates of the \( j \)th keypoint from the \( i \)th person.
Figure 3: The overall architecture of InsPose. $C_3$, $C_4$, $C_5$ are the feature maps of the backbone network (e.g., ResNet-50). $P_3$ to $P_7$ are the FPN feature maps as in [21, 35], which are used for final predictions. $H \times W$ denotes the height and width of feature maps, and $K$ is the number of keypoints for each instance. Classification head is used to classify the locations on the feature maps into "person" or "not person". The parameters of KP-Nets are dynamically generated with respect to different locations of the input. Then the learned parameters are split and reshaped as the weights and biases of convolutions in each KP-Net which is applied to predict keypoint maps for the particular instance. Disk offset branch predicts the instance-agnostic local offset to recover the discretization error caused by down-sampling.

Multi-Person Pose Representation. In this work, our core idea is that for an image with $N$ instances, $N$ different KP-Nets will be dynamically generated, and each KP-Net will contain the characteristics of the target instance in its filter kernels. As shown in Figure 1, when the KP-Net for person $i$ is applied to an input, it will produce a keypoint map $M_i \in \mathbb{R}^{H \times W \times K}$, where $s$ is the down-sampling ratio of the map. Then the maximal activation in $j$th channel of $M_i$ (denoted as $M_{ij}$) indicates the location of the $j$th keypoint for person $i$, which can be formulated as

$$\hat{P}^j_i = \left( \tilde{x}^j_i, \tilde{y}^j_i \right) = \arg \max_{x, y} M_{ij}$$

(2)

To recover the discretization error caused by the down-sampling of keypoint maps, we additionally propose a disk offset prediction module. It predicts an instance-agnostic local offset $O \in \mathbb{R}^{H \times W \times 2K}$ for each category of keypoints. Then we pick the offset vector $(\Delta x, \Delta y)$ corresponding to the above keypoint location $\hat{P}^j_i$, the final keypoint location on the input image plane can be defined as

$$\tilde{P}^j_i = \left( \left( \tilde{x}^j_i + \Delta x \right) \times s, \left( \tilde{y}^j_i + \Delta y \right) \times s \right)$$

(3)

Network Architecture. We build the proposed InsPose on the popular anchor-free object detector FCOS [35] due to its simplicity and flexibility, as shown in Figure 3. We make use of ResNet [13] or HRNet [32] as the backbone network. Then feature pyramid networks (FPN) [21] is employed to produce 5 feature pyramid levels $[P_3, P_4, P_5, P_6, P_7]$, whose down-sampling ratios are 8, 16, 32, 64, 128, respectively.

As shown in previous works [12, 30], each location on the FPN’s feature maps is considered as a positive sample if it is associated with a ground-truth instance. Otherwise the location is regarded as a negative sample (i.e., background). Specifically, suppose that the feature maps with a down-sampling ratio of $s$ can be represented as $P_i \in \mathbb{R}^{H/2^s \times W/2^s \times C}$. A location $(x, y)$ on the feature maps $P_i$ can be mapped back onto the input image as $\left( x/2^s + l_2^x, y/2^s + l_2^y \right)$. As described in previous works [34, 35], if the mapped location falls in the center region of an instance, the location is assigned to be responsible for that instance. The center region is defined by the box $(c_x - r, c_y - r, c_x + r, c_y + r)$, where $(c_x, c_y)$ denotes the pseudo-box (minimum enclosing rectangles of the keypoints) center of the instance. $r$ denotes the radius of positive sample regions and is a constant being 1.5 as in FCOS [35].
and conv3 \((8 \times 17 + 17)\). As mentioned before, the dynamically generated KP-Net contains information about the instance at the location. Thus, it will ideally only fire on the pixels of the instance’s keypoints, as shown in Figure 1.

**Shared Inputs.** As shown in Figure 3, there is a keypoint feature branch connected to FPN level \(P_3\) with a down-sampling ratio of 8, which provides the shared feature map \(F_{kp} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times C_{kp}}\) (i.e., \(s = 8\)) for the final pose estimation. It first stacks four \(3 \times 3\) convolutions with 128 channels before the last layer. Then, to reduce the number of parameters in the KP-Net, the last layer of the keypoint feature branch decreases the number of channels from 128 to 8 (i.e., \(C_{kp} = 8\)). Considering that the relative coordinate map can provide a strong cue for distinguishing different instances, we therefore combine \(F_{kp}\) with a relative coordinate map \(R_{x,y} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times 2}\), which are relative coordinates to the location \((x, y)\) (i.e., where the parameters \(\theta_{x,y}\) are generated). Afterwards, the combination \(\tilde{F}_{x,y} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times (C_{kp} + 2)}\) is fed into the KP-Nets, which is omitted in Figure 3 for simplicity.

**Optimization.** The loss function of the instance-aware KP-Nets module can be formulated as:

\[
L_{kp} \left( \{ \theta_{x,y} \} \right) = \frac{1}{N_{pos}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}^* > 0\}} L_{ce} \left( \text{FCN} \left( \tilde{F}_{x,y}; \theta_{x,y} \right), M_{x,y}^* \right),
\]

where FCN denotes the fully convolutional dynamic networks, which consists of the generated instance-aware parameters \(\theta_{x,y}\) at location \((x, y)\). The classification label \(c_{x,y}^*\) of location \((x, y)\) is 1 if the location is associated with a ground-truth person and is 0 for indicating background. \(\mathbb{1}_{\{c_{x,y}^* > 0\}}\) is the indicator function, being 1 if \(c_{x,y}^* > 0\) and 0 otherwise. \(N_{pos}\) is the total number of locations where \(c_{x,y}^* > 0\). As described before, \(\tilde{F}_{x,y}\) is the combination of the keypoint feature \(F_{kp}\) and the relative coordinate map \(R_{x,y}\).

As mentioned before, the resolution of the predicted keypoint map is \(\frac{1}{s}\) of the input image’s resolution. Thus, for each visible ground-truth keypoint of an instance, we compute its low-resolution equivalent \(\hat{p} = \left( \lfloor \frac{x}{s} \rfloor, \lfloor \frac{y}{s} \rfloor \right)\). The training target is a one-hot \(\frac{H}{s} \times \frac{W}{s}\) binary mask where only a single pixel at location \(\hat{p}\) is labeled as foreground. And \(M_{x,y}^* \in \{0, 1\}^{\frac{H}{s} \times \frac{W}{s} \times K}\) represents the \(K\) one-hot masks of the person instance associated with location \((x, y)\). \(L_{ce}\) is the softmax cross-entropy loss, which encourages a single keypoint to be detected. These operations are omitted in Equation (4) for clarification.

### 3.3 Disk Offset Estimation

Suppose that \(M_i \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times K}\) is the predicted keypoint map for the \(i\)th person. The location of maximal activation in the \(j\)th channel of \(M_i\) indicates the location of the \(j\)th keypoint of person \(i\), which is defined as \(\hat{p}_j^i = (x_j^i, y_j^i)\). We can get the final keypoint location by mapping back \(\hat{p}_j^i\) onto input image plane as \((S X_j^i + 0.5) \times s, (S Y_j^i + 0.5) \times s\). However, there is a discretization error in this naive position mapping due to the down-sampling operation. In order to compensate for the discretization error, we predict an instance-agnostic local offset \(O \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times 2K}\) for each category of keypoints.
As shown in Figure 3, the disk offset branch takes the FPN feature maps \( P_3 \) as input. Afterwards, three \( 3 \times 3 \) conv layers are applied for the final disk offset prediction. There are \( 2K \) output channels in the offset prediction, indicating the displacements in both the horizontal and vertical directions. Let \( p \) be the \( 2D \) position in the feature plane, and \( D_R = \{ p : ||p - P_j|| \leq R \} \) be a disk of radius \( R \) centered around \( P_j \), where \( P_j \) represents the \( 2D \) position of the \( j \)-th keypoint of the \( i \)-th person. As illustrated in Figure 4, for each position \( p \) within the disk region \( D_R \), the \( 2D \) offset vector which points from the image position \( x_j \) to keypoint \( P_j \) is generated. During training, we penalize the disk offset prediction errors with the \( L_1 \) loss. The supervision is only performed in the disk region. During inference, the position of maximum response \((\bar{x}_i, \bar{y}_i)\) in the keypoint map is shifted by the corresponding offset vector \( \Delta(\bar{x}, \bar{y}) \). Then the final result can be formulated as \( (\Delta(\bar{x}, \bar{y}) \times x, \Delta(\bar{x}, \bar{y}) \times y) \). This well-designed module can further improve pose estimation performance, as shown in our experiments.

### 3.4 Training and Inference

**Training and Loss Function.** Given the fact that multi-task learning facilitates the training of models, we adopt a multi-person heatmap prediction as an auxiliary task, which is omitted in Figure 3 for simplicity. The heatmap prediction task takes as input the FPN feature maps \( P_3 \). Afterwards, two \( 3 \times 3 \) convolutions with channel being 256 are applied here. Finally, another \( 3 \times 3 \) convolution layer with the output channel being \( K \) is appended for the final heatmap prediction, where \( K \) is the number of keypoint categories in the dataset. As shown in the following experiments, this joint learning procedure can boost the performance of our model. Note that the heatmap-based task is only used for auxiliary supervision during training and is removed when testing.

Formally, the overall loss function of InsPose can be formulated as:

\[
L = L_{cls} + L_{kp} + L_{do} + L_{hm}
\]

where \( L_{cls} \) is the focal loss function [22] for the classification branch, \( L_{kp} \) is the loss of instance-aware KP-Nets module as defined in Equation (4), \( L_{do} \) is \( L_1 \) loss for disk offset regression, \( L_{hm} \) is a variant of focal loss [19] for the auxiliary heatmap branch.

**Inference.** The inference of InsPose is simple and straightforward. Given an input image, we forward it through the network to obtain the outputs including classification confidence \( p_{x,y} \) and the generated parameters \( \theta_{x,y} \) of the KP-Net at location \((x, y)\). We first use a confidence threshold of 0.1 to filter out predictions with low confidence. Then we select the top \( N \) (i.e., \( N = 500 \)) scoring person instances and their corresponding parameters. Then the \( N \) groups of pose estimation results can be obtained through Equations (2) and (3). At last, We do non-maximum suppression (NMS) using the minimum enclosing rectangles of estimated keypoints of the instances. The top 100 instance keypoints in each image are kept for evaluation.

It is worth noting that our framework can conveniently be adapted to simultaneous bounding-box and keypoint detection by attaching a bounding-box head, which shares the preceding convolutions of the parameter generating branch for KP-Nets.

### 4 EXPERIMENTS

We evaluate our proposed InsPose on the large-scale COCO benchmark dataset [23], which contains more than 250K instances where each person has 17 annotated keypoints. Consistent with previous methods [11, 33, 39], we use the COCO train2017 split (57K images) for training and val2017 split (5K images) as validation for our ablation study. The main results are reported on the test-dev split (20K images) for comparison with other methods. We use the standard COCO metrics including \( AP, AP_{50}, AP_{75}, AP_M, AP_L \) to evaluate the performance of pose estimation.

**Implementation Details:** Unless specified, ablation studies are conducted with ResNet-50 [13] and FPN [21]. The models are trained with stochastic gradient descent (SGD) with a minibatch of 16 images. The initial learning rate is set to 0.01. Weight decay and momentum are set as 0.0001 and 0.9, respectively. Specifically, the model is trained for 36 epochs and the initial learning rate is divided by 10 at epoch 27 and 33 in ablation experiments. For the main results on test-dev split, we train the models for 100 epochs and the initial learning rate is divided by 10 at epoch 75 and 90. ImageNet [8] pre-trained ResNet [13] is employed to initialize our backbone networks. We initialize the newly added layers as described in [22]. When training, the images are resized to have their shorter sides in \([640, 800]\) and their longer sides less or equal to 1333, and then they are randomly horizontally flipped with the probability of 0.5. No extra data augmentation like rotation [26] or random cropping [33] is used during training. When testing, the input images are resized to have their shorter sides being 800 and their longer sides less or equal to 1333.

#### 4.1 Ablation Experiments

**Baseline.** First, we conduct the experiments through employing instance-aware dynamic networks to produce keypoint maps where the maximal activation of each channel is chosen as the keypoint. As shown in Table 1 (1st row), the naive framework can obtain competitive performance (60.3% in \( AP_{kp} \)). As shown in Figure 3, we apply the local offset predicted by the disk offset branch to recover the discretization error of keypoint maps. As shown in Table 1 (2nd row), the performance is improved from 60.3% to 62.1% in

| Disk offset | Heatmap | AP  | AP\(_{50}\) | AP\(_{75}\) | AP\(_M\) | AP\(_L\) | AR  | AR\(_{50}\) | AR\(_{75}\) |
|-------------|---------|-----|----------|----------|--------|--------|-----|----------|----------|
| Our InsPose |         | 60.3| 85.4     | 65.5     | 55.0   | 68.4   | 68.6| 91.0     | 73.7     |
| Our InsPose | ✓       | 62.1| 86.0     | 67.5     | 57.6   | 69.1   | 70.2| 91.4     | 75.3     |
| Our InsPose | ✓ ✓     | 63.1| 86.2     | 68.5     | 58.5   | 70.1   | 70.9| 91.2     | 76.1     |
AP\textsuperscript{kp}, which demonstrate the effectiveness of disk offset branch. Therefore, in the sequel, we will use the disk offset branch for all the following experiments. By jointly learning with a heatmap prediction task, the performance of our model can be further improved from 62.1% to 63.1%, as shown in Table 1 (3rd row). Note that the heatmap prediction is only used during training to provide the multi-task regularization and is pruned during inference. We show some visualization results of InsPose in Figure 2.

**Architecture of KP-Net.** In this section, we discuss the architecture of the KP-Nets in InsPose. Our baseline is the KP-Net of three $1 \times 1$ convolutions with 8 channels considering the high capacity of conditional convolutions, which contains a total of 313 parameters. As shown in Table 2 (3rd row), it achieves 63.1% in AP\textsubscript{kp}. Then, we conduct experiments by varying the depth of the KP-Net. As shown in Table 2, the depth being 1 achieves inferior performance as the KP-Net is actually equivalent to a linear mapping, which has overly weak capacity in this case. With the depth increases, the performance is improved gradually due to the higher capacity of the KP-Net. However, the performance drops slightly when the depth increases further mainly due to the reason that it is harder to optimize more dynamic parameters.

**Resolution of Keypoint Maps and Disk Offset Predictions.** Intuitively, the resolution of the above two predictions is of great importance to the final performance due to pose estimation being a position-sensitive task. We investigate this in the experiment. As shown in Figure 3, the resolution of the origin keypoint score maps and disk offset predictions is same as the resolution of P3 features, which is $\frac{1}{16}$ of the input image resolution. As shown in Table 3, our baseline model (2nd row in the table) achieves 63.1% in AP\textsubscript{kp}. If both of the two predictions is downsampled to the $\frac{1}{16}$ of the input image resolution (1st row in the table), the performance drops slightly (from 63.1% to 62.6%). In particular, AP\textsubscript{M} drops remarkably by 1.3% (from 58.5% to 57.2%), which indicating that low-resolution predictions harms the performance of small-scale persons. Additionally, upsampling both of the predictions by a factor of 2 and 4 (using bilinear interpolation), the performance is degraded by 1.1% and 1.6%, respectively. This is probably due to the relatively low-quality annotations of keypoints, which causes a growing negative impact on the training of the model as the output resolution increases.

**Combining with Bounding Box Detection.** By simply attaching a bounding-box regression head, we can simultaneously detect bounding-boxes and keypoints of the instances. When testing, NMS is applied to the detected bounding-boxes instead of the minimum enclosing rectangles of keypoints. Here we confirm it by the experiment. As shown in Table 4, our framework can achieve a reasonable person detection performance, which is competitive with the Faster R-CNN detector in Mask R-CNN [11] (52.5% vs. 50.7%). Meanwhile, the keypoint performance drops a little due to different positive/negative label assignments between the detection-boxes and pseudo-boxes at training phase.

### 4.2 Comparisons with State-of-the-art Methods

In this section, we evaluate the proposed single-stage multi-person pose estimator on MS-COCO test-dev and compare it with other state-of-the-art methods. As shown in Table 5, without any bells and whistles (e.g., multi-scale and flipping testing, the refining in [3, 25], and any other tricks), the proposed framework achieves 65.4% and 66.3% in AP\textsubscript{kp} on COCO test-dev split, with ResNet-50 and ResNet-101 as the backbone, respectively. With multi-scale testing, our framework can reach 71.0% in AP\textsubscript{kp} with HRNet-w32.

**Compared to Single-stage Methods:** Our method significantly outperforms existing single-stage methods, such as DirectPose [33], CenterNet [43], SPM [26] and Point-Set Anchors [39]. The performance of our method is +3.2% and +3.0% higher as compared to DirectPose [33] with ResNet-50 and ResNet-101, respectively. InsPose with HRNet-w32 even outperforms the latest single-stage pose estimator Point-Set Anchors [39] with HRNet-w48, 71.0% vs. 68.7% in AP\textsubscript{kp}. Note that our model is anchor-free and Point-Set Anchors need 27 carefully chosen pose anchors per location.

**Compared to Bottom-up Methods:** Our method outperforms the state-of-the-art bottom-up methods, such as CMU-Pose [3], AE [25], PersonLab [27], PifPaf [18], HigherHRNet [6] and the latest DGCN [29]. With single-scale testing, it achieves significant improvement over HigherHRNet, 69.3% vs. 66.4% in AP\textsubscript{kp}, using the same backbone HRNet-w32. Besides that, our performance with

---

**Table 2:** Ablation experiments on COCO val2017 with different depth of KP-Nets. "depth": the number of conv. layers in the KP-Nets.

| depth | AP  | AP\textsubscript{50} | AP\textsubscript{75} | AP\textsubscript{M} | AP\textsubscript{L} |
|-------|-----|---------------------|---------------------|-------------------|-------------------|
| 1     | 54.7 | 81.1               | 57.4               | 50.5              | 62.2              |
| 2     | 62.4 | 85.9               | 67.9               | 58.0              | 69.2              |
| 3     | 63.1 | 86.2               | 68.5               | 58.5              | 70.1              |
| 4     | 61.7 | 85.9               | 67.2               | 57.1              | 68.7              |

**Table 3:** Ablation experiments on COCO val2017 with different resolution of keypoint score maps and disk offset predictions. "res. ratio": the resolution ratio of the above two predictions to the input image.

| res. ratio | AP  | AP\textsubscript{50} | AP\textsubscript{75} | AP\textsubscript{M} | AP\textsubscript{L} |
|-----------|-----|---------------------|---------------------|-------------------|-------------------|
| 1/16      | 62.6 | 85.8               | 68.2               | 57.2              | 70.5              |
| 1/8       | 63.1 | 86.2               | 68.5               | 58.5              | 70.1              |
| 1/4       | 62.0 | 83.9               | 67.5               | 57.8              | 69.1              |
| 1/2       | 45.4 | 73.6               | 47.0               | 39.1              | 54.7              |

**Table 4:** Ablation experiments on COCO val2017 for InsPose with person bounding-box detection. The proposed framework can achieve reasonable person detection results (52.5% in box AP). As a reference, the Faster R-CNN person detector in Mask R-CNN [11] achieves 50.7% in box AP.

| w/ bbox | AP\textsubscript{bb} | AP\textsubscript{kp} | AP\textsubscript{kp}\textsubscript{50} | AP\textsubscript{kp}\textsubscript{75} | AP\textsubscript{kp}\textsubscript{M} | AP\textsubscript{kp}\textsubscript{L} |
|---------|----------------------|----------------------|-------------------------------------|-----------------------------------|----------------------------------|----------------------|
| -       | 63.1                 | 86.2                 | 68.5                               | 58.5                             | 70.1                            | 70.3                 |
Table 5: Comparison with state-of-the-art methods on MS COCO test-dev dataset. ∗ and † denote multi-scale testing and using additional refinement, respectively. We compare the average inference time per image of our methods with other state-of-the-art single-stage and two-stage methods. The time is counted with single-scale testing on a single NVIDIA TITAN X GPU.

| Method          | Backbone          | AP  | AP50 | AP75 | AP_M | AP_L | Times[s] |
|-----------------|-------------------|-----|------|------|------|------|----------|
| **Top-down Methods** |
| Mask-RCNN [11]  | ResNet-50         | 62.7| 87.0 | 68.4 | 57.4 | 71.1 | 0.08     |
| CPN [5]         | ResNet-Inception  | 72.1| 91.4 | 80.0 | 68.7 | 77.2 | > 0.65   |
| HRNet [32] †    | HRNet-w32         | 74.9| 92.5 | 82.8 | 71.3 | 80.9 | > 2.34   |
| ECSI [31]       | ResNet-152        | 74.3| 91.8 | 81.9 | 70.7 | 80.2 | -        |
| RSN [2] ∗       | 2 × RSN-50        | 75.5| 93.6 | 84.0 | 73.0 | 79.6 | > 3.06   |
| **Bottom-up Methods** |
| CMU-Pose∗ † [3] | 3CM-3PAF (102)    | 61.8| 84.9 | 67.5 | 57.1 | 68.2 | -        |
| AE∗ † [25]      | Hourglass-4 stacked | 65.5| 87.1 | 71.4 | 61.3 | 71.5 | 0.52     |
| PersonLab [27]  | ResNet-101        | 65.5| 87.1 | 71.4 | 61.3 | 71.5 | -        |
| PifPaf [18]     | ResNet-152        | 66.7| -    | -    | 62.4 | 72.9 | 0.26     |
| HigherHRNet [6] | HRNet-w32         | 66.4| 87.5 | 72.8 | 61.2 | 74.2 | 0.60     |
| DGCN [29]       | ResNet-152        | 67.4| 88.0 | 74.4 | 63.6 | 73.0 | > 0.26   |
| **Single-stage Methods** |
| DirectPose [33] | ResNet-50         | 62.2| 86.4 | 68.2 | 56.7 | 69.8 | -        |
| DirectPose [33] | ResNet-101        | 63.3| 86.7 | 69.4 | 57.8 | 71.2 | -        |
| CenterNet [43]  | Hourglass-104     | 63.0| 86.8 | 69.6 | 58.9 | 70.4 | 0.16     |
| SPM∗ † [26]     | Hourglass-8 stacked | 66.9| 88.5 | 72.9 | 62.6 | 73.1 | -        |
| Point-Set Anchors∗ [39] | HRNet-w48 | 68.7| 89.9 | 76.3 | 64.8 | 75.3 | 0.27     |
| Our InsPose     | ResNet-50         | 65.4| 88.9 | 71.7 | 60.2 | 72.7 | 0.08     |
| Our InsPose ∗   | ResNet-50         | 67.1| 89.7 | 74.0 | 62.6 | 73.8 | -        |
| Our InsPose ∗   | ResNet-101        | 66.3| 89.2 | 73.0 | 61.2 | 73.9 | 0.10     |
| Our InsPose ∗   | ResNet-101        | 67.8| 90.3 | 74.9 | 64.3 | 73.4 | -        |
| Our InsPose ∗   | HRNet-w32         | 69.3| 90.3 | 76.0 | 64.8 | 76.1 | 0.15     |
| Our InsPose ∗   | HRNet-w32         | 71.0| 91.3 | 78.0 | 67.5 | 76.5 | -        |

HRNet-w32 is higher than DGCN [29] with ResNet-152, 69.3% vs. 67.4% in AP<sub>kp</sub>.

**Compared to Top-down Methods:** With the same backbone ResNet-50, the proposed method outperforms previous strong baseline Mask R-CNN (65.4% vs. 62.7% in AP<sub>kp</sub>). Our model is still behind other top-down methods, since they mitigate the scale variance of persons by cropping and resizing the detected person into the same scale. As noted before, two-stage methods take an additional person detector and employ a single person pose estimator to each cropped image, resulting in high computation overhead. In contrast, our method is much simpler and faster since we eliminate the redundant person detector.

**Inference Time Comparison:** We measure the inference time of all variants of our method and other methods on the same hardware if possible. As shown in Table 5, InsPose with HRNet-w32 can infer faster than Point-Set Anchors [39] (0.15s vs. 0.27s per image). And it also runs significantly faster than state-of-the-art top-down methods, such as HRNet [32] (>2.34s per image) and RSN [2] (>3.06s per image). Note that the time of additional person detector used in HRNet and RSN is not included here. Moreover, we notice that the grouping post-processing in the bottom-up method HigherHRNet [6] (0.60s per image) is time-consuming and results in inferior speed compared to InsPose. Overall, InsPose makes a good trade-off between the accuracy and efficiency, benefiting from its simple single-stage framework.

5 CONCLUSION

In this paper, we present a single-stage multi-person pose estimation method, termed InsPose. It directly maps a raw input image to the desired instance-aware poses, get rid of the need for the grouping post-processing in bottom-up methods or the bounding-box detection in top-down approaches. Specifically, we propose an instance-aware module to adaptively adjust (part of) the network parameters for each instance. Our method can significantly increase the capacity and adaptive-ability of the network for recognizing various poses, while maintaining a compact end-to-end trainable pipeline. We also propose a disk offset branch to recover the discretization error due to down-sampling, boosting the keypoint detection performance further. We conduct extensive experiments on the MS-COCO dataset where our method achieves significant improvement over existing single-stage methods and performs comparably with state-of-the-art two-stage methods in terms of accuracy and efficiency.
