End-to-End Multi-Person Pose Estimation with Transformers

Dahu Shi[1*] Xing Wei[2*] Liangqi Li[1] Ye Ren[1] Wenming Tan[1†]
[1]Hikvision Research Institute, Hangzhou, China
[2]School of Software Engineering, Xi'an Jiaotong University
{shidahu, liliangqi, renye, tanwenming}@hikvision.com, weixing@mail.xjtu.edu.cn

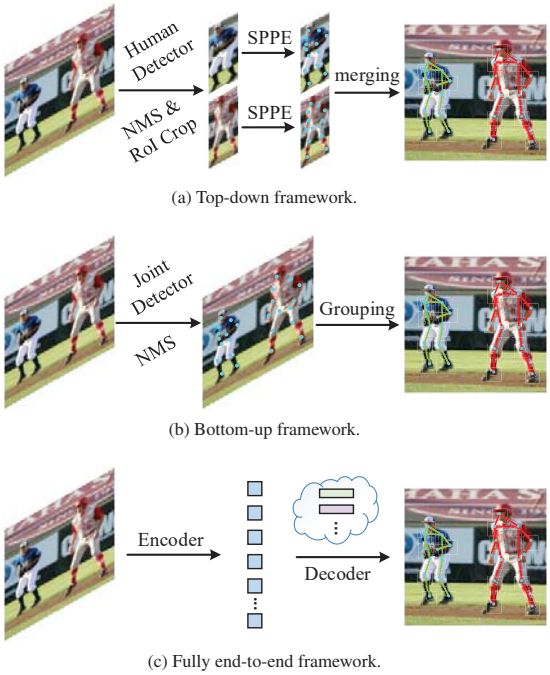
Abstract

Current methods of multi-person pose estimation typically treat the localization and association of body joints separately. In this paper, we propose the first fully end-to-end multi-person Pose Estimation framework with TRansformers, termed PETR. Our method views pose estimation as a hierarchical set prediction problem and effectively removes the need for many hand-crafted modules like RoI cropping, NMS and grouping post-processing. In PETR, multiple pose queries are learned to directly reason a set of full-body poses. Then a joint decoder is utilized to further refine the poses by exploring the kinematic relations between body joints. With the attention mechanism, the proposed method is able to adaptively attend to the features most relevant to target keypoints, which largely overcomes the feature misalignment difficulty in pose estimation and improves the performance considerably. Extensive experiments on the MS COCO and CrowdPose benchmarks show that PETR plays favorably against state-of-the-art approaches in terms of both accuracy and efficiency. The code and models are available at https://github.com/hikvision-research/opera.

1. Introduction

Multi-person pose estimation (*aka*, keypoint detection) aims to detect all the instances and identify the kinematic joints of each person simultaneously. It is one of the fundamental computer vision tasks and has a wide range of applications such as action recognition [9], human-computer interaction [15], pedestrian tracking [1, 31] and re-identification [22], *etc*.

Existing mainstream methods solve this challenging task with two-stage frameworks, including top-down and bottom-up approaches. Top-down methods [5, 10, 12, 33, 39], as illustrated in Figure 1a, first detect each individual by a person detector and then transfer the task to a simpler single-person pose estimation problem. The top-down pipeline comes with the following drawbacks: 1) the pose estimation accuracy heavily relies on the performance of person detector, incurring inferior performance in complex scenarios [7]; 2) the computational cost is expensive due to the use of the isolated detector [26, 32] and the running time depends on the number of instances in the image. On the other hand, bottom-up methods [3, 17, 27, 30] (shown in Figure 1b) first detect all potential keypoints in the image in an instance-agnostic manner, and then perform a grouping post-processing to get instance-aware full-body poses. The grouping process is usually heuristic, hand-crafted and



(a) Top-down framework.

(b) Bottom-up framework.

(c) Fully end-to-end framework.

Figure 1. **Comparison of mainstream pose estimation frameworks.** SPPE in (a) indicates single-person pose estimation. We proposed a fully end-to-end framework as show in (c).

*Co-first authors. †Corresponding author.

Framework		RoI-free	Grouping-free	NMS-free
Two-stage	Top-down		✓	
	Bottom-up	✓		
Single-stage	Non end-to-end	✓	✓	
	Fully end-to-end	✓	✓	✓

Table 1. **Comparison of pose estimation frameworks.**

cumbersome [27], involving several hyper-parameters and tricks. These kinds of methods split the pose estimation problem into two steps and are often not optimized in a fully end-to-end fashion.

Recently, there has been a great interest to directly estimate multi-person poses from the input image in a single stage [26, 29, 32, 34, 36, 41]. SPM [29] propose a structured pose representation that unifies person instance and body joint position representations and simplifies the multi-person pose estimation pipeline. FCPose [26] and InsPose [32] propose a fully convolutional multi-person pose estimation framework using dynamic instance-aware convolutions, which is compact and efficient. These methods eliminate the need for RoI (Region of Interest) cropping and keypoint grouping post-processing and achieve a good trade-off between accuracy and efficiency. However, they still rely on "taking-peak" on the heatmap [29, 41] or score map [26, 32] and hand-crafted NMS (Non-Maximum Suppression) post-processing [26, 32, 36], which are still not end-to-end optimized.

Inspired by the paradigm emerged in object detection [4, 42], we present a fully end-to-end multi-person pose estimation framework (Sec. 3.1) with transformers, termed PETR. The proposed method unifies person instance and fine-grained body joint localization by formulating pose estimation as a hierarchical set prediction problem. Given multiple randomly initialized pose queries, a pose decoder (Sec. 3.3) learns to reason about the relations of objects [14] and estimate a set of instance-aware poses under the global image context. Then, a joint decoder (Sec. 3.4) is designed to explore the structured relations between different joints and further optimize the full-body poses at a finer level. Compared with existing single-stage methods, PETR could hierarchically attend to the features most relevant to target keypoints, largely overcomes the feature misalignment issue [11, 34] and improves the performance considerably. Our end-to-end query-based framework is learned via the bipartite matching strategy that avoids the heuristic label assignment and eliminates the need for NMS post-processing.

We illustrate and compare the mainstream pose estimation frameworks in Figure 1 and Table 1. The main contributions of this work are summarized as follows.

- We propose the first fully end-to-end learning framework for multi-person pose estimation. The proposed PETR method directly predicts instance-aware full-body poses and eliminates the need for RoI cropping, grouping, and NMS post-processings.

- We design hierarchical decoders to deal with the feature misalignment issue, and capture both relations between person instances and kinematic joints by the attention mechanism.

- PETR surpasses all single-stage and bottom-up methods and is comparable to top-down methods on COCO dataset. Besides, PETR performs well in crowded scenes and establishes a new state of the art on CrowdPose dataset.

2. Related Work

2.1. Multi-Person Pose Estimation

The existing multi-person pose estimation approaches can be summarized into three categories: top-down methods, bottom-up methods and recent single-stage methods.

Top-down methods. The top-down methods first employ an object detector to obtain the bounding box of each person instance in an image. Then the instance is cropped from the bounding box for single-person pose estimation. Representative works include Hourglass [28], RMPE [10], CPN [5], SimpleBaseline [39], HRNet [33] and so on. In general, top-down methods have a slow inference speed. They break the multi-person pose estimation task into two steps: person detection and single-person pose estimation. Instead of cropping RoIs from the original image, Mask R-CNN [12] utilizes RoIAlign operation to extract features of RoIs from the feature maps of the detector, significantly speeding up the inference. Moreover, top-down methods are highly dependent on the performance of the detector.

Bottom-up methods. The bottom-up methods detect all keypoints in an instance-agnostic fashion, and then group them into individuals. Most existing bottom-up methods mainly focus on how to associate the detected keypoints that belong to the same person. OpenPose [3] utilizes part affinity fields to establish connections between keypoints of the same instance. Associative embedding [27] produces a detection heatmap and a tagging map for each body joint, and then groups keypoints with similar tags into an individual. PersonLab [30] groups keypoints by directly learning a 2D offset field for each pair of keypoints. PifPaf [17] learns a Part Association Field (PAF) to connect the keypoints into full-body poses. Compared to top-down methods, 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.

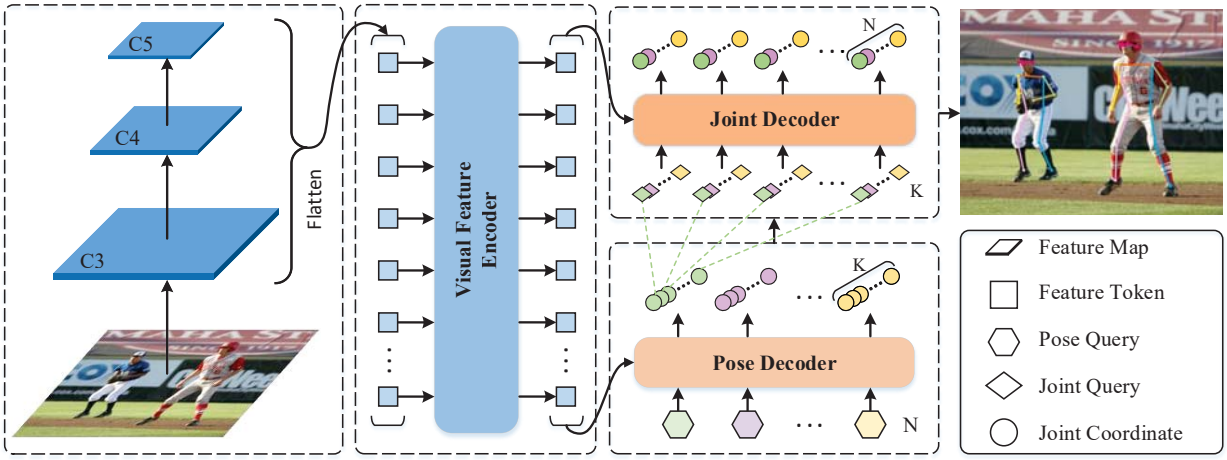


Figure 2. **The overall architecture of PETR**. C3 to C5 are multi-scale feature maps extracted from the backbone network (*e.g.*, ResNet-50). The visual feature encoder takes the flattened image features as inputs and refines them. Given N pose queries and the refined multi-scale feature tokens, pose decoder predicts N full-body poses in parallel. After that, an additional joint decoder takes each scattered pose (*i.e.*, kinematic joints of each pose) as its reference points and outputs the refined pose as final results. K is the number of keypoints for each instance (*e.g.*, $K = 17$ in COCO [21] dataset).

Single-stage methods. To avoid the aforementioned limitations in both top-down and bottom-up methods, the single-stage methods [26, 29, 32, 34, 36, 41] are proposed to densely regress a set of pose candidates over spatial locations, where each candidate consists of the keypoint positions that are from the same person. SPM [29] proposes a structured pose representation to unify position information of person instances and body joints. Due to the weak regression results, CenterNet [41] proposes to match the regressed keypoint positions to the closest keypoints detected from the keypoint heatmaps. Point-set anchors [36] adopt deformable-like convolutions to refine the predefined pose anchors, mitigating the difficulties of feature misalignment. FCPose [26] and InsPose [32] utilize dynamic instance-aware convolutions to solve the multi-person pose estimation problem, achieving better accuracy/efficiency trade-off than other single-stage methods. Although these approaches obtain competitive performance, they are not fully end-to-end optimized and still need heuristic post-processing like NMS or keypoint location correction [41].

2.2. Transformer in Vision

Transformer [35] has been widely applied in natural language processing. Recently, many works attempted to involve transformer architecture in computer vision tasks and showed promising performances [4, 6, 8, 37, 42]. ViT [8] apply the transformer to encode a sequence of image patches for image classification. DETR [4] and Deformable DETR [42] adopt transformer architecture together with bipartite matching to perform object detection in an end-to-end fashion. MaskFormer [6] and SOIT [40] employ transformer decoders to predict a set of binary masks directly, and effectively remove the need for many hand-crafted components. SAANet [37] proposes a scene-adaptive transformer network for crowd counting, achieving highest accuracy on several benchmarks. PRTR [20] and TFPose [25] formulate the pose estimation task as a regression problem by transformers. However, they still follow the top-down framework and need the hand-crafted RoI cropping operation. In this paper, we use transformer to build a fully end-to-end framework for multi-person pose estimation.

3. Methodology

3.1. Overall Architecture

As depicted in Figure 2, the proposed framework consists of three key modules: visual feature encoder, pose decoder and joint decoder, where (1) the visual feature encoder is applied to refine the multi-scale feature maps extracted from the backbone network, (2) the pose decoder is employed to predict multiple full-body poses, and (3) the joint decoder is designed to further refine the full-body poses at a joint level.

Given an image $I \in \mathbb{R}^{H \times W \times 3}$, we extract multi-scale feature maps C_3, C_4 and C_5 from the last three stages of the backbone (*e.g.*, ResNet [13]), whose strides are 8, 16 and 32, respectively. The multi-scale feature maps are projected to the ones with 256 channels by a spatial-wise fully-connected (FC) layer and then flattened into feature tokens C'_3, C'_4 and C'_5. Specifically, the shape of C'_i is $L_i \times 256$,

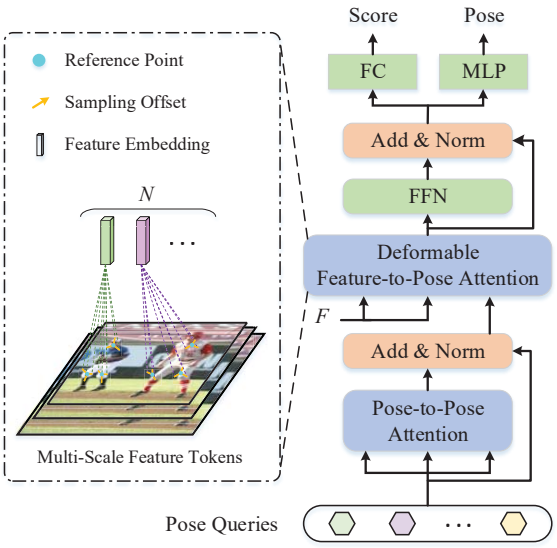


Figure 3. **Detailed structure of the pose decoder.** Given N pose queries, the pose decoder outputs N instance-aware full-body poses. The progressive deformable cross-attention module can attend to the visual features most relevant to the target keypoints.

where $L_i = \frac{H}{2^i} \times \frac{W}{2^i}$. Next, using the concatenated feature tokens $[C_3', C_4', C_5']$ as input, the visual feature encoder outputs the refined multi-scale feature tokens $F \in \mathbb{R}^{L \times 256}$, where $L = L_3 + L_4 + L_5$ is the total number of feature tokens. After that, N randomly initialized pose queries are utilized to directly reason N full-body poses (and their corresponding confidence score) under the global image context. Finally, we scatter each full-body pose into a sequence of body joints and adopt a joint decoder to further refine them.

3.2. Visual Feature Encoder

High-resolution and multi-scale feature maps are important for the pose estimation task [7, 33]. Since the multi-head self-attention module [4, 8] has quadratic computation complexity to input size, we employ the deformable attention module [42] to implement our feature encoder.

Due to the low computational complexity of the deformable attention layer, our encoder can merge and refine the multi-scale feature maps. Concretely, each encoder layer comprises a multi-scale deformable attention module and a feed-forward network (FFN). In order to identify which feature level each feature token lies in, we add a scale-level embedding, in addition to the positional embedding. There are six deformable encoder layers stacked in sequence in our visual feature encoder. After that, we can obtain the refined multi-scale visual feature memory F.

3.3. Pose Decoder

In the pose decoder, we aim to reason a set of full-body poses under the global image context (*i.e.*, feature memory F). Similar to the visual feature encoder, we use the deformable attention module to build our pose decoder due to its efficiency. Specifically, given N randomly initialized pose queries $Q_{pose} \in \mathbb{R}^{N \times D}$, the pose decoder outputs N full-body poses $\{\mathcal{P}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times 2K}$, where $\mathcal{P}_i = \{(x_i^j, y_i^j)\}_{j=1}^{K}$ denotes the coordinates of K joints for the i^{th} person and D indicates the dimension of the query embedding.

The detailed structure of the pose decoder is illustrated in Figure 3. First, the query embeddings are fed into the self-attention module for interacting with each other (*i.e.*, pose-to-pose attention). Then each query extracts features from the multi-scale feature memory F via the deformable cross-attention module (*i.e.*, feature-to-pose attention). There are K reference points, serving as the initial locations of a full-body pose in our deformable cross-attention module, in contrast to [42]. Subsequently, the instance-aware query features are fed into the multi-task prediction heads. The classification head predicts the confidence score for each object by a linear projection layer (FC). The pose regression head predicts the relative offsets w.r.t. the K reference points using a multi-layer perceptron (MLP) with a hidden size of 256. There are three decoder layers applied sequentially in our pose decoder.

Instead of only using the final decoder layer to predict the pose coordinates, inspired by [42], we leverage all the decoder layers to estimate the pose coordinates progressively. Specifically, each layer refines the poses based on the predictions from the previous layer. Formally, given a normalized pose $\mathcal{P}_{d-1}$ predicted by the $(d-1)^{th}$ decoder layer, the d^{th} decoder layer refines the pose as

$$\mathcal{P}_d = \sigma(\sigma^{-1}(\mathcal{P}_{d-1}) + \Delta\mathcal{P}_d), \tag{1}$$

where $\Delta\mathcal{P}_d$ are predicted offsets at the d^{th} layer, σ and σ^{-1} denote the `sigmoid` and `inverse sigmoid` function, respectively. In this way, $\mathcal{P}_{d-1}$ serves as the new reference point of cross-attention module in the d^{th} decoder layer. The initial reference point $\mathcal{P}_0$ is a randomly-initialized matrix and jointly updated with the model parameters during training. As a result, the progressive deformable cross-attention module can attend to the visual features most relevant to the target keypoints, which overcoming the feature misalignment issue naturally.

3.4. Joint Decoder

As shown in Figure 4, the joint decoder is proposed to explore the structured relations between articulated joints and further refine full-body poses at a joint level. We employ deformable attention module to build our joint decoder

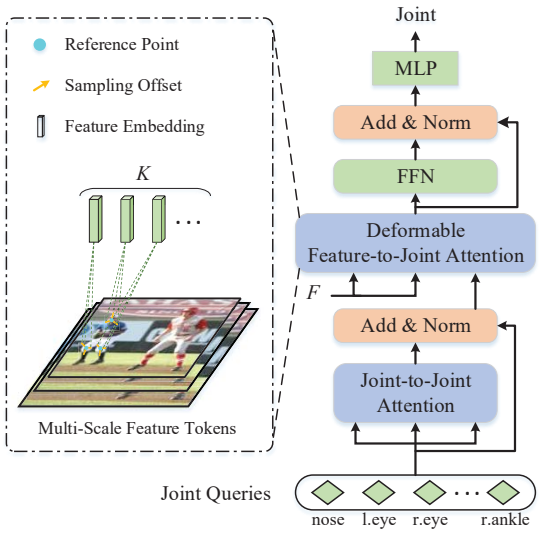


Figure 4. **Detailed structure of the joint decoder.** Each one of the K joint queries takes a keypoint location of the full-body pose predicted by the pose decoder as its reference point for further refinement.

as in the pose decoder. Concretely, given K randomly initialized joint queries $Q_{joint} \in \mathbb{R}^{K\times D}$, the joint decoder takes the joint locations of each full-body pose predicted by preceding pose decoder as their initial reference points and then further refine the joint locations. Note that all the poses can be processed in parallel since they are independent of each other in the joint decoder.

The detailed structure of the joint decoder is illustrated in Figure 4. The joint queries firstly interact with each other via a self-attention module (*i.e.*, joint-to-joint attention), and then extract visual features in a deformable cross-attention module (*i.e.*, feature-to-joint attention). Subsequently, a joint regression head predicts the 2-D joint displacement $\Delta J = (\Delta x, \Delta y)$ by applying an MLP. Similar to the pose decoder, the joint coordinates are progressively refined. Formally, let J_{d-1} be the normalized joint coordinates predicted by the $(d-1)^{th}$ decoder layer, the predictions of the d^{th} decoder layer are $J_d = \sigma(\sigma^{-1}(J_{d-1}) + \Delta J_d)$, where J_0 is joint locations of the pose predicted by the proceeding pose decoder.

3.5. Loss Functions

Following [4], we use a set-based Hungarian loss that forces a unique prediction for each ground-truth pose. The same classification loss function (denoted as L_{cls}) as in [42] is used for classification head in our pose decoder. Besides, we adopt both L_1 loss (denoted as L_{reg}) and OKS loss (denoted as L_{oks}) for pose regression head and joint regression head in our pose decoder and joint decoder, respectively.

OKS loss. The most commonly-used L_1 loss have different scales for small and large poses even if their relative errors are similar. To mitigate this issue, we propose to use the Object Keypoint Similarity (OKS) loss additionally, which can be formulated as,

$$L_{oks}(P, P^*) = \frac{\sum_i^K \exp(-\|P_i - P_i^*\|/2s^2k_i^2)\delta(v_i > 0)}{\sum_i^K \delta(v_i > 0)}, \tag{2}$$

where $\|P_i - P_i^*\|$ is the Euclidian distance between the i^{th} predicted keypoint and ground-truth one, v_i is the visibility flag of the ground truth, s is the object scale, and k_i is a per-keypoint constant that controls falloff. As shown above, the OKS Loss is normalized by the scale of the person instance with the importance of keypoints equalized.

Heatmap loss. Similar to [26,32], we use the auxiliary heatmap regression training for fast convergence. We gather the feature tokens from C_3 outputs of visual feature encoder and reshape the tokens into the original spatial shape. The result is denoted by $F_{C_3} \in \mathbb{R}^{(H/8)\times(W/8)\times D}$. We apply a deformable transformer encoder to generate the heatmap prediction. Then, we compute a variant of focal loss [18] between the predicted and ground-truth heatmaps (denoted as L_{hm}). Note that the heatmap branch is only used for aided training and is discarded in inference.

Overall loss. Formally, the overall loss function of our model can be formulated as:

$$L = L_{cls} + \lambda_1 L_{reg} + \lambda_2 L_{oks} + \lambda_3 L_{hm} \tag{3}$$

where λ_1, λ_2 and λ_3 are the loss weights, respectively.

4. Experiments

4.1. COCO Keypoint Detection

We evaluate the performance on the COCO dataset [21], which contains over $200K$ images and $250K$ person instances labeled with 17 keypoints. All the models are trained on the `train2017` set ($57K$ images). We use the `val2017` set ($5K$ images) as validation for our ablation experiments and compare with other state-of-the-art methods on the `test-dev` set ($20K$ images).

Evaluation metrics. The standard evaluation metric is based on Object keypoint Similarity (OKS). We report standard average precision and recall scores[1]: AP^{50} (AP at OKS $= 0.50$), AP^{75}, AP (mean of AP scores from OKS $= 0.50$ to OKS $= 0.95$ with the increment as 0.05), AP^M for persons of medium sizes and AP^L for persons of large sizes.

Training details. Following the setting of [26, 32], we augment the input image by random crop, random flip, and random resize (the shorter sides in [480,800] and the longer

[1] http://cocodataset.org/#keypoints-eval

	Method	Backbone	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	Time [ms]
			Two-stage methods					
Top-down	Mask R-CNN [12]	ResNet-50	62.7	87.0	68.4	57.4	71.1	89
	Mask R-CNN*	ResNet-50	63.9	87.7	69.9	59.7	71.5	89
	Mask R-CNN*	ResNet-101	64.3	88.2	70.6	60.1	71.9	108
	CPN [5]	ResNet-Inception	72.1	91.4	80.0	68.7	77.2	>472
	SimpleBaseline† [39]	ResNet-152	73.7	91.9	81.1	70.3	80.0	>784
	PRTR [20]	HRNet-w32	72.1	90.4	79.6	68.1	79.0	-
	HRNet† [33]	HRNet-w32	74.9	92.5	82.8	71.3	80.9	>632
	HRNet† [33]	HRNet-w48	75.5	92.5	83.3	71.9	81.5	>857
Bottom-up	CMU-Pose‡ [3]	3CM-3PAF	61.8	84.9	67.5	57.1	68.2	-
	CMU-Pose [2]	VGG-19	64.2	86.2	70.1	61.0	68.8	74
	AE† [27]	Hourglass-4 stacked	62.8	84.6	69.2	57.5	70.6	139
	PifPaf [17]	ResNet-152	66.7	-	-	62.4	72.9	260
	HrHRNet† [7]	HRNet-w32	66.4	87.5	72.8	61.2	74.2	400
	DEKR† [11]	HRNet-w32	67.3	87.9	74.1	61.5	76.1	411
	SWAHR† [24]	HRNet-w32	67.9	88.9	74.5	62.4	75.5	406
			Single-stage methods					
Non end-to-end	DirectPose [34]	ResNet-50	62.2	86.4	68.2	56.7	69.8	74
	FCPose [26]	ResNet-50	64.3	87.3	71.0	61.6	70.5	68
	InsPose [32]	ResNet-50	65.4	88.9	71.7	60.2	72.7	80
	DirectPose [34]	ResNet-101	63.3	86.7	69.4	57.8	71.2	-
	FCPose [26]	ResNet-101	65.6	87.9	72.6	62.1	72.3	93
	InsPose [32]	ResNet-101	66.3	89.2	73.0	61.2	73.9	100
	CenterNet [41]	Hourglass-104	63.0	86.8	69.6	58.9	70.4	160
	Point-Set Anchors†‡ [36]	HRNet-w48	68.7	89.9	76.3	64.8	75.3	-
Fully end-to-end	PETR (Ours)	ResNet-50	67.6	89.8	75.3	61.6	76.0	89
	PETR‡ (Ours)	ResNet-50	69.2	90.5	77.1	64.2	76.4	-
	PETR (Ours)	ResNet-101	68.5	90.3	76.5	62.5	77.0	95
	PETR‡ (Ours)	ResNet-101	70.0	90.9	78.2	65.3	77.1	-
	PETR (Ours)	Swin-L	70.5	91.5	78.7	65.2	78.0	133
	PETR‡ (Ours)	Swin-L	71.2	91.4	79.6	66.9	78.0	-

Table 2. **Comparisons with state-of-the-art methods on COCO `test-dev` dataset.** † and ‡ denote flipping and multi-scale test, respectively. Mask R-CNN* are the results from `Detectron2` [38], which are better than the original results reported in the Mask R-CNN paper [12]. We measure the inference time of other methods on the same hardware if possible and all the times are counted with single-scale test. Note that some top-down methods need extra inference time of person detector which is not contained in this table.

sides less or equal to 1333). The models are trained with Adam optimizer [16] with base learning rate of 2×10^{-4}, momentum of 0.9 and weight decay of 1×10^{-4}. Specifically, we train the model for 50 epochs with a total batch size of 32 and the initial learning rate is decayed at 40^{th} epoch by a factor of 0.1 in ablation experiments. For the main results on `test-dev` set, the model is trained for 100 epochs and the initial learning rate is decayed at 80^{th} epoch by a factor of 0.1.

Testing details. The input images are resized to have their shorter sides being 800 and their longer sides less or equal to 1333. For the multi-scale test, we resize the original images with their short sides being 800, 1000, and 1200 respectively. All reported numbers have been obtained with single model without model ensemble. The inference time is measured using a single NVIDIA Tesla V100 GPU.

4.2. Results on COCO test-dev

We firstly make comparisons with the state-of-the-art methods, as shown in Table 2. When using the same backbone network as the feature extractor, our PETR outperforms all existing bottom-up methods as well as the single-stage methods with or without multi-scale test. Without any bells and whistles, the proposed method achieves 67.6 and 68.5 AP scores, with ResNet-50 and ResNet-101 as the backbone, respectively. Our best model with Swin-L [23] achieves **71.2** AP score on COCO `test-dev2017`.

Comparison with single-stage methods. Our method significantly outperforms existing single-stage methods, such as DirectPose [34], CenterNet [41], Point-Set Anchors [36] and InsPose [32]. The performance of our method is 2.2 points higher compared with InsPose [32] with both ResNet-50 and ResNet-101 as the backbone. Our PETR

Figure 5. **Visualization results of PETR**. The first row and the second row show the visualization results on COCO `val2017` and CrowdPose test set, respectively. PETR performs well on a wide range of poses, containing viewpoint change, occlusion, motion blur and crowded scene. Best viewed in color.

with ResNet-101 even outperforms Points-Set Anchors with HRNet-w48, which has a much larger size than ResNet-101, recording 70.0 $vs.$ 68.7 in AP score. Note that our approach is NMS-free which make it more efficient compared with these single-stage methods.

Comparison with two-stage methods. With a more compact pipeline, we even outperforms the state-of-the-art bottom-up methods, such as CMU-Pose [2], AE [27], PifPaf [17], HigherHRNet [7], DEKR [11] and SWAHR [24]. With single-scale test, PETR achieves significantly improvement over HigherHRNet [7], 68.5 $vs.$ 64.7 in AP score, in which our PETR using a smaller backbone ResNet-101 than HRNet-w32 used in HigherHRNet [7]. Our method also outperforms the latest proposed SWAHR [24], 68.5 $vs.$ 67.9, with a smaller backbone. Moreover, PETR outperforms previous strong baseline Mask R-CNN [12] with backbone ResNet-101 (68.5 $vs.$ 64.3 in AP), while maintain a competitive inference speed.

Comparison of inference time. We measure the inference time of our models with different backbones and other methods on the same hardware if possible. As shown in Table 2, PETR with ResNet-50 could achieve competitive inference speed to the typical top-down method, Mask R-CNN [12], and the single-stage method InsPose [32], *i.e.* 89ms $vs.$ 89ms $vs.$ 80ms. We also show the speed-accuracy trad-off between our PETR and state-of-the-art methods in Figure 6, and PETR surpasses all those bottom-up methods in both speed and accuracy field. Although it seems a little slower than some of the other methods (FCPose [26]), we should note that current computational devices like GPU are not specifically optimized for the transformer-based architecture.

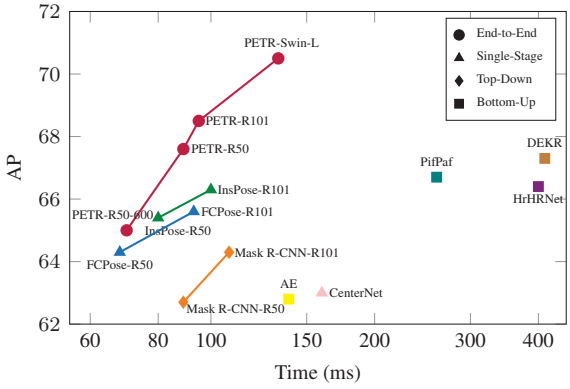


Figure 6. **The speed-accuracy trade-off comparison.** PETR-R50-600 indicates a variant of PETR with ResNet-50 backbone where the short side of the input image is 600 pixels.

4.3. Ablation Study

We perform a number of ablation experiments to analyze effectiveness of the proposed pose/joint decoders and OKS loss on the COCO `val2017` dataset.

Pose and joint decoders. PETR use hierarchical decoders (*i.e.*, the pose decoder and joint decoder) to regress keypoint locations progressively. The pose decoder alone already estimates full-body poses, which could be refined by the joint decoder further. As shown in Table 3, the joint decoder improves the AP by 1.0 points. Note that the improvement of AP^{75} is more significant (1.3 points), indicating a finer prediction offered by the joint decoder. Moreover, we conduct another experiment where both the pose

Pose decoder	Joint decoder	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
		49.4	75.7	55.1	46.4	54.3	60.4
✓		66.4	**87.2**	73.6	60.6	74.8	73.7
✓	✓	**67.4**	87.0	**74.9**	**61.7**	**75.9**	**74.8**

Table 3. **Ablation experiments:** ablation of the proposed pose decoder and joint decoder on COCO `val2017`. The first row means that the refined multi-scale feature tokens are directly utilized to regress full-body poses, which incurs severe feature misalignment as mentioned in [34].

OKS loss	OKS matching	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
		64.2	86.7	70.1	58.4	73.5	73.9
✓		65.6	87.7	72.1	60.3	73.9	74.8
	✓	66.9	86.6	74.4	60.9	75.8	74.5
✓	✓	**67.4**	**87.0**	**74.9**	**61.7**	**75.9**	**74.8**

Table 4. **Ablation experiments:** the effect of the OKS loss and its matching cost on COCO `val2017`.

Method	AP	AP^{50}	AP^{75}	AP^{E}	AP^{M}	AP^{H}
Top-down methods						
Mask R-CNN [12]	57.2	83.5	60.3	69.4	57.9	45.8
AlphaPose [10]	61.0	81.3	66.0	71.2	61.4	51.1
SimpleBaseline [39]	60.8	81.4	65.7	71.4	61.2	51.2
SPPE [19]	66.0	84.2	71.5	75.5	66.3	57.4
Bottom-up methods						
OpenPose [3]	-	-	-	62.7	48.7	32.3
HrHRNet† [7]	65.9	86.4	70.6	73.3	66.5	57.9
DEKR† [11]	67.3	86.4	72.2	74.6	68.1	58.7
SWAHR† [24]	71.6	88.5	77.6	**78.9**	72.4	63.0
Fully end-to-end methods						
PETR (Ours)	71.6	90.4	78.3	77.3	72.0	**65.8**
PETR† (Ours)	**72.0**	**90.9**	**78.8**	78.0	**72.5**	65.4

Table 5. **Comparisons with state-of-the-art methods on CrowdPose test dataset.** Superscripts E, M, H of AP stand for easy, medium and hard images, respectively. † denotes flipping test.

decoder and joint decoder are disabled. In this case, we only utilize the multi-scale feature tokens refined by the visual feature encoder to regress full-body poses directly. The performance (1st row in Table 3) drops remarkably due to the misalignment between features and target joints, as mentioned in [11, 34].

OKS loss and OKS matching cost. Following DETR [4], we use a bipartite matching mechanism to indicate the relationship between the training samples and ground truths, and then compute several types of loss to supervise the model. OKS is a commonly-used evaluation metric in pose estimation benchmarks. However, most methods use L_1 loss for training, therefore leave a gap between optimizing the loss and maximizing the OKS metric. To our knowledge, this is the first work to adopt OKS as the loss function in the pose estimation field. We conduct experiments to study the impact of OKS loss and its matching cost, respectively. As shown in Table 4, the OKS loss brings 1.4 AP score improvement and using OKS for matching cost gains 2.7 AP score. When combining both two components, the performance is significantly improved from 64.2 to 67.4.

4.4. CrowdPose

We further evaluate our approach on the CrowdPose [19] dataset that is more challenging and includes many crowded scenes. It consists of $20K$ images, containing about 80,000 persons. Each person is labeled with 14 body joints. The train, val and test datasets contain about $10K$, $2K$ and $8K$ images, respectively. We train our models on the train and val sets and report the results on the test set as done in [7].

Evaluation metrics. The standard average precision based on OKS which is the same as COCO is adopted as the evaluation metrics. The CrowdPose dataset is split into three crowding levels: easy, medium and hard. We report the following metrics: AP, AP^{50}, AP^{75}, as well as AP^{E}, AP^{M} and AP^{H} for easy, mudium and hard images.

Test set results. The results of our approach and other state-of-the-art methods on the test set are shown in Table 5. Different from the top-down methods which have lost their superiority in crowded scenes, our approach shows its robustness and achieves 72.0 AP score, which surpasses the latest bottom-up method SWAHR [24], especially on AP^{H} item. Our PETR does not depend on detection results like top-down methods, and does not need NMS to suppress redundant results like bottom-up and other single-stage methods, which makes it more flexible and suitable to estimate human pose under the crowded scenes.

5. Conclusion

This paper presents the first fully end-to-end multi-person pose estimation framework, termed PETR. It reformulates multi-person pose estimation as a hierarchical set prediction problem, which effectively removes the need for many hand-crafted components like RoI cropping, grouping, and NMS post-processings. PETR is simple and direct, offering a better trade-off between accuracy and efficiency than other methods.

Acknowledgments

This work is funded by National Natural Science Foundation of China under Grant No. 62006183, National Key Research and Development Project of China under Grant No. 2020AAA0105600, China Postdoctoral Science Foundation under Grant No. 2020M683489, and the Fundamental Research Funds for the Central Universities under Grant No. xhj032021017-04 and xzy012020013.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5167–5176, 2018. 1

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 6, 7

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 1, 2, 6, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 3, 4, 5, 8

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 1, 2, 6

[6] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv preprint arXiv:2107.06278*, 2021. 3

[7] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020. 1, 4, 6, 7, 8

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3, 4

[9] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 1

[10] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2334–2343, 2017. 1, 2, 8

[11] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14676–14686, 2021. 2, 6, 7, 8

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 6, 7, 8

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[14] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018. 2

[15] Himanshu Prakash Jain, Anbumani Subramanian, Sukhendu Das, and Anurag Mittal. Real-time upper-body human pose estimation using a depth camera. In *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, pages 227–238. Springer, 2011. 1

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 9, 2015. 6

[17] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019. 1, 2, 6, 7

[18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 5

[19] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10863–10872, 2019. 8

[20] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1944–1953, 2021. 3, 6

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 5

[22] Jinxian Liu, Bingbing Ni, Yichao Yan, Peng Zhou, Shuo Cheng, and Jianguo Hu. Pose transferrable person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4099–4108, 2018. 1

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021. 6

[24] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13264–13273, 2021. 6, 7, 8

[25] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021. 3

[26] Weian Mao, Zhi Tian, Xinlong Wang, and Chunhua Shen. Fcpose: Fully convolutional multi-person pose estimation with dynamic instance-aware convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9034–9043, 2021. 1, 2, 3, 5, 6, 7

[27] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2, 6, 7

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2

[29] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6951–6960, 2019. 2, 3

[30] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018. 1, 2

[31] Deva Ramanan, David A Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 271–278. IEEE, 2005. 1

[32] Dahu Shi, Xing Wei, Xiaodong Yu, Wenming Tan, Ye Ren, and Shiliang Pu. Inspose: Instance-aware networks for single-stage multi-person pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3079–3087, 2021. 1, 2, 3, 5, 6, 7

[33] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 1, 2, 4, 6

[34] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *arXiv preprint arXiv:1911.07451*, 2019. 2, 3, 6, 8

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[36] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *European Conference on Computer Vision*, pages 527–544. Springer, 2020. 2, 3, 6

[37] Xing Wei, Yuanrui Kang, Jihao Yang, Yunfeng Qiu, Dahu Shi, Wenming Tan, and Yihong Gong. Scene-adaptive attention network for crowd counting. *arXiv preprint arXiv:2112.15509*, 2021. 3

[38] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[39] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 1, 2, 6, 8

[40] Xiaodong Yu, Dahu Shi, Xing Wei, Ye Ren, Tingqun Ye, and Wenming Tan. Soit: Segmenting objects with instance-aware transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 3

[41] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 3, 6

[42] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021. 2, 3, 4, 5