Group Pose: A Simple Baseline for End-to-End Multi-person Pose Estimation

Huan Liu[1,3*] Qiang Chen[2*] Zichang Tan[2] Jiang-Jiang Liu[2] Jian Wang[2] Xiangbo Su[2]
Xiaolong Li[1,3] Kun Yao[2] Junyu Han[2] Errui Ding[2] Yao Zhao[1,3†] Jingdong Wang[2]
[1]Institute of Information Science, Beijing Jiaotong University [2]Baidu VIS
[3]Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China

Abstract

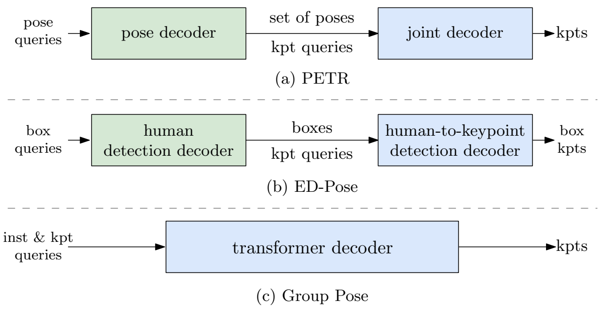
In this paper, we study the problem of end-to-end multi-person pose estimation. State-of-the-art solutions adopt the DETR-like framework, and mainly develop the complex decoder, e.g., *regarding pose estimation as keypoint box detection and combining with human detection in ED-Pose [38], hierarchically predicting with pose decoder and joint (keypoint) decoder in PETR [27].*

We present a simple yet effective transformer approach, named Group Pose. We simply regard K-keypoint pose estimation as predicting a set of $N \times K$ keypoint positions, each from a keypoint query, as well as representing each pose with an instance query for scoring N pose predictions.

Motivated by the intuition that the interaction, among across-instance queries of different types, is not directly helpful, we make a simple modification to decoder self-attention. We replace single self-attention over all the $N \times (K + 1)$ queries with two subsequent group self-attentions: (i) N within-instance self-attention, with each over K keypoint queries and one instance query, and (ii) $(K+1)$ same-type across-instance self-attention, each over N queries of the same type. The resulting decoder removes the interaction among across-instance type-different queries, easing the optimization and thus improving the performance. Experimental results on MS COCO and CrowdPose show that our approach without human box supervision is superior to previous methods with complex decoders, and even is slightly better than ED-Pose that uses human box supervision. Paddle [1] and PyTorch [2] codes are available.

1. Introduction

Multi-person pose estimation aims to detect the corresponding human keypoints for all human instances in an image. Previous frameworks include top-down [25, 6, 35, 4, 29, 33] and bottom-up methods [1, 24, 12, 5] that divide the task into two sequential sub-tasks: human detection with single-person pose estimation or human-agnostic keypoint detection with human instance grouping [24]. Another line in previous frameworks is one-stage methods [22, 26, 31, 34], which directly predict instance-aware keypoints. These frameworks rely on non-differentiable hand-crafted post-processes [10, 24], which complicate the pipelines and challenge the optimizations. Inspired by the success of DETR [2] in object detection, building an end-to-end framework for multi-person pose estimation has seen significant interest.



(a) PETR

(b) ED-Pose

(c) Group Pose

Figure 1: **Comparison of transformer decoders.** Here we mainly illustrate the overview of the decoder part of PETR [27], ED-Pose [38], and our Group Pose. The three end-to-end frameworks differ in query design and decoder architecture. Group Pose only uses a simple transformer decoder rather than developing complex decoders. 'inst' and 'kpt' represent for instance and keypoint.

Recent approaches follow the DETR framework [2, 41, 39], with the transformer encoder-decoder architecture for multi-person pose estimation, as shown in Figure 1. PETR [27] hierarchically predicts the keypoint positions and uses two subsequent decoders, pose decoder and joint decoder, with two different queries, pose query (a person has one pose query) for pose decoder, and keypoint queries for joint decoder. ED-Pose [38] transfers pose estimation to a keypoint box detection problem, and learns a content query and a box query for each keypoint position prediction

[*] Equal Contribution. Work done when H. Liu is an intern at Baidu.
[†] Corresponding author. Email: yzhao@bjtu.edu.cn
[1] Code of Paddle: https://github.com/Michel-liu/GroupPose-Paddle
[2] Code of PyTorch: https://github.com/Michel-liu/GroupPose

with using the box size to process the query.

In this paper, we present a simple yet effective transformer approach, named Group Pose, for end-to-end multi-person pose estimation. Instead of using a single query to predict and score one pose, similar to ED-Pose, we use the $N \times K$ keypoint queries to regress the $N \times K$ positions, by regarding each keypoint as an object, as well as N instance queries, each representing a K-keypoint pose for scoring the K-keypoint pose prediction.

We make a simple modification for the decoder architecture. We replace standard self-attention in the decoder with two subsequent group self-attentions: N parallel self-attentions with each over K keypoint queries and the corresponding instance query for exploiting kinematic relation and gathering information for scoring pose predictions, and $(K+1)$ parallel self-attentions with each over N queries of the same type for collecting duplicate prediction information like self-attention of the original DETR.

The two self-attentions capture two kinds of interactions: (i) N within-instance interactions over K keypoint queries and one instance query, (ii) $(K+1)$ across-instance interactions over N queries of the same type (*e.g.*, nose keypoint query or instance query). The extra interactions in standard self-attention are (iii) across-instance interactions for queries with different types, which is not directly useful. Empirical results show that the removal of the third kind of interactions eases the optimization and thus improves the performance.

The design about self-attention is different from the closely-related approach ED-Pose [38]. On the one hand, in addition to removing the third interactions, ED-Pose only models across-instance interactions for the instance queries. In contrast, our approach also models across-instance interactions for the K keypoint types. On the other hand, our approach separates the two interactions using two subsequent group self-attentions, explicitly exploring the information about queries belonging to the same human instance, and with the same type. ED-Pose couples the two interactions using a single masked self-attention with the third interactions masked. It is empirically demonstrated that the two differences benefit the pose estimation performance.

Experimental results show that our simple approach Group Pose without human box supervision surprisingly outperforms the recent end-to-end methods with human box supervisions on MS COCO [16] and CrowdPose [14]. Notably, Group Pose achieves 72.0 AP with ResNet-50 [9] and 74.8 AP with Swin-Large [18] on MS COCO `val2017`. We hope our simple transformer decoder in Group Pose will motivate people to simplify the design in end-to-end multi-person pose estimation.

2. Related Work

Multi-person pose estimation is a challenging task that aims to detect the corresponding human keypoints for all human instances in an image. Previous methods usually adopt complex frameworks to address it, which are divided into non-end-to-end and end-to-end methods.

Non-end-to-end methods. There are typically two types: two-stage and one-stage. Two-stage frameworks, including top-down [25, 6, 35, 4, 29, 33] and bottom-up methods [1, 24, 12, 5], split the multi-person pose estimation task into two sequential sub-tasks, human detection with single-person pose estimation or human-agnostic keypoint detection with human instance grouping. In top-down methods, an object detector is first employed to detect the boxes of human instances, which is then cropped for single-person pose estimation in each box. Bottom-up methods first predict all human keypoints in a human-agnostic way then group them into instances. While one-stage frameworks [40, 22, 26, 31, 34] directly predict instance-aware keypoints. These methods require hand-crafted pose-processes, such as NMS [10] or grouping [24], which complicate the pipelines and challenge the optimizations. In this paper, we focus on concise ways, end-to-end frameworks.

End-to-end methods. Current end-to-end multi-person pose estimation frameworks [27, 36, 38] are built by following the designs of DETR [2] and its variants [41, 23, 3, 17, 13, 39]. They adopt a paradigm of splitting the multi-person pose estimation task into two sub-processes. For example, PETR [27] views the task as a hierarchical set prediction problem. It first determines human instances by predicting a set of poses with a pose decoder and then refines the keypoints in each pose with a joint (keypoint) decoder. There are also two different types of queries, pose query (a person has one pose query) for pose decoder and keypoint queries for joint decoder. QueryPose [36] and ED-Pose [38] follow this end-to-end paradigm but further incorporate an extra human detection task. QueryPose [36] follows Sparse R-CNN [30] to build two parallel RoIAlign-based [8] decoders to perform human detection and pose estimation, respectively. ED-Pose [38] transfers pose estimation task to a keypoint box detection problem. It first employs a human detection decoder [39] to determine the human instances with box queries. Then it builds a human-to-keypoint detection decoder with a content query and a box query for each keypoint position, collecting contextual information near keypoint positions.

Although these end-to-end methods show promising results in multi-person pose estimation, they rely on complex decoders. Our Group Pose, on the other hand, adopts a simple transformer decoder, improving the performance and simplifying the process.

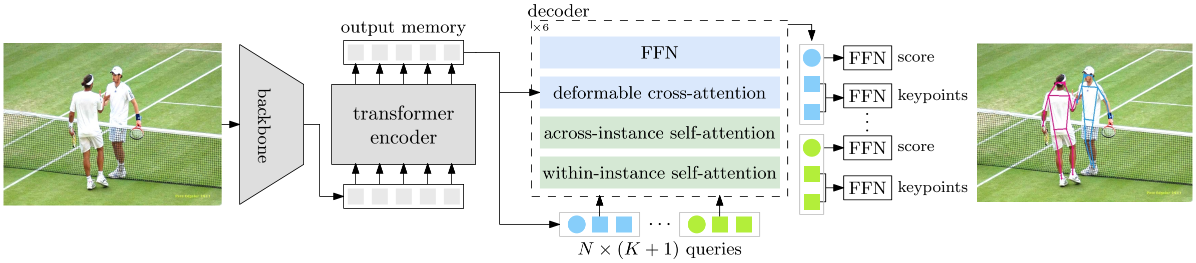


Figure 2: **Our Group Pose architecture.** The backbone takes an image as input and outputs image features, which are refined by the transformer encoder. To directly predict N human poses with K keypoint positions in each pose, we adopt $N \times (K+1)$ queries, containing N instance queries for scoring poses and $N \times K$ keypoint queries for regressing positions.

3. Group Pose

Group Pose is a simple yet effective end-to-end multi-person pose estimation framework. We follow previous end-to-end frameworks [27, 38] to view the multi-person pose estimation task as a set prediction problem, but directly adopt a simple transformer decoder [41] instead of complex decoders, simplifying the process. Next, we introduce the key elements of Group Pose.

3.1. Overview

The overall structure of Group Pose is depicted in Figure 2. Group Pose consists of a backbone [9, 18], a transformer encoder [32], a transformer decoder, and task-specific prediction heads. This framework enables Group Pose to simultaneously regresses K keypoints (*e.g.*, $K = 17$ on MS COCO) for N human instances given an image.

Backbone and transformer encoder. We directly follow DETR frameworks [41] to build the backbone and the transformer encoder (with 6 deformable transformer layers [41]) for Group Pose. It takes an image as input and outputs the extracted multi-level features, which serve as inputs for the following transformer decoder. We use 4 feature levels in Group Pose, with downsampling rates of $\{8, 16, 32, 64\}$.

Transformer decoder. For transformer decoder, we adopt a combination of $N \times K$ keypoint queries and N instance queries as input instead of using a single query to predict and score one pose. The keypoint queries regard each keypoint as an object and are used to regress the $N \times K$ keypoint positions, while each instance query is for scoring the corresponding K-keypoint pose prediction. Besides, the architecture of transformer decoder is simple, which stacks 6 same decoder layers [41]. In each decoder layer, we follow the macro design of previous DETR frameworks by building self-attention, cross-attention implemented with deformable attention, and FFN. We only make simple modifications to self-attention in our Group Pose. Specifically, we replace the standard self-attention with two subsequent group self-attentions, enabling decoder layers to perform interactions over queries belonging to the same human instance and with the same type.

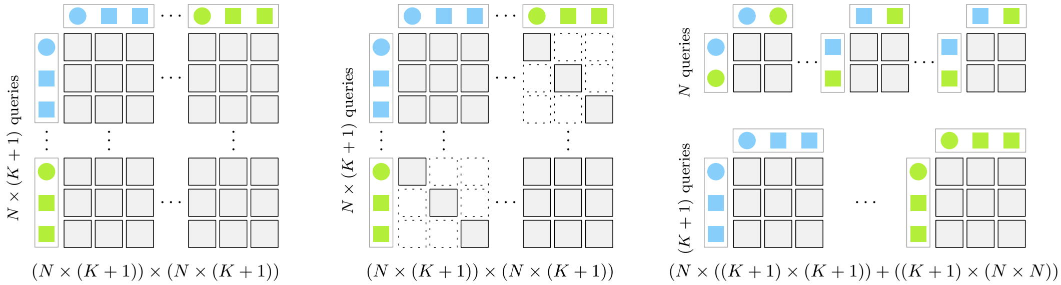
Prediction heads. There are two prediction heads implemented with FFNs in Group Pose for human classification and human keypoints regression. Group Pose predicts for N human poses, each contains a classification score and K keypoint positions for the corresponding K-keypoint pose.

Loss function. The Hungarian matching algorithm [2] is employed for one-to-one assignment between predicted poses and ground-truth poses. Our loss function comprises solely of classification loss ($\mathcal{L}_{cls}$) and keypoint regression loss ($\mathcal{L}_{kpt}$), without any extra supervisions such as human detection loss in QueryPose [36] and ED-Pose [38] or heatmap loss in PETR [27]. The keypoint regression loss ($\mathcal{L}_{kpt}$) is a combination of a normal ℓ_1 loss and a constrained ℓ_1 loss named Object Keypoint Similarity (OKS) [27]. We directly use the cost coefficients and loss weights of ED-Pose [38] in the hungarian matching and the calculation of losses.

3.2. $N \times K$ keypoint queries and N instance queries

In multi-person pose estimation, frameworks are required to predict N human poses with K keypoint positions in each pose given an image. We directly use $N \times K$ keypoints queries to predict the poses, with each K predicted keypoint positions to represent a corresponding K-keypoint pose for a human instance. To classify if it is a human instance by scoring the predicted N human poses, we also introduce N instance queries. Thus, in Group Pose, a human instance can be represented with a combination of K keypoint queries and one instance query. As these two types of queries are responsible for different tasks, we construct and initialize them differently.

Query construction. We follow the previous end-to-end frameworks [27, 38] to first identify human instances and predict human poses in each position of the output memory from the transformer encoder. Then we select N human instances ($N = 100$) based on the classification scores, resulting N human poses and the corresponding output memory features of the selected N positions (with the shape of $N \times D$, where D is the channel dimension). We then construct and initialize the keypoint queries and the instance


$N \times (K+1)$ queries
$(N \times (K+1)) \times (N \times (K+1))$
$N \times (K+1)$ queries
$(N \times (K+1)) \times (N \times (K+1))$
N queries
$(K+1)$ queries
$(N \times ((K+1) \times (K+1)) + ((K+1) \times (N \times N))$

(a) standard self-attention (b) standard self-attention with mask (c) group self-attentions

Figure 3: **Conceptual comparison of self-attention implementations.** Given $N \times (K+1)$ queries, three different implementations of self-attention are considered in the decoder layers. (a) standard self-attention over all queries. (b) standard self-attention with an attention mask, removing the across-instance interactions for queries with different types. (c) our proposed group self-attentions. Dashed boxes indicate the masked-out interactions.

queries based on the above results.

For $N \times K$ keypoint queries, we construct the content part of K keypoint queries ($K \times D$) in each human instance by combining K randomly initialized learnable keypoint embeddings ($K \times D$) and the corresponding output memory feature ($1 \times D$). And the position part of K keypoint queries ($K \times 2$) in each human instance is initialized with the corresponding predicted K-keypoint pose ($K \times 2$). For N instance queries, we only consider the content part and use a randomly initialized learnable instance embedding ($1 \times D$) for each human instance. This is because instance query is for the classification task, which does not requires explicit position information. When performing cross-attention in the decoder layers, we simply use the mean of K keypoint positions as the reference point for the instance query in each human instance.

3.3. Group self-attentions

In DETR frameworks [2, 41, 23], self-attention in the transformer decoder is usually applied to model interactions among queries, collecting information from other queries and facilitating the duplicate removal for instances. While the situation is slightly different in Group Pose, as there two different types of queries, K keypoint queries and one instance query for each human instance. Motivated by the intuition that the interactions among across-instance queries of different types, *e.g.*, the interactions between the instance query and the keypoint queries across human instances, may not be directly helpful for the above purposes. We thus replace the standard self-attention over all the $N \times (K+1)$ queries with two subsequent group self-attentions: N within-instance self-attentions and $K+1$ same-type across-instance self-attentions.

N within-instance self-attentions. We build interactions among the queries within a human instance, exploiting kinematic relations and gathering information for scoring pose predictions. It calculates N self-attention maps for N human instances in parallel with shapes of $(K+1) \times (K+1)$, omitting the dimensions of batch size and attention heads.

$(K+1)$ same-type across-instance self-attentions. Similar to the within-instance group self-attentions, We add another group self-attentions, which collect information from the same-type queries in other instances and help remove duplicate predictions. We build interactions across human instances with the same-type queries, the instance query and each keypoint queries, resulting in $K+1$ same-type across-instance self-attentions. It also can be implemented in parallel with one self-attention module, calculating $K+1$ self-attention maps with shapes of $N \times N$.

Compared with the standard self-attention, our two subsequent group self-attentions explicitly explores the information about queries belong to the same human instance and with the same type, while remove the across-instance interactions for queries with different types. Empirical results in Section 4.3 show that the removal of this kind of interactions eases the optimization and thus improves the performance for our Group Pose.

4. Experiments

4.1. Settings

Datasets. Our experiments are conducted on two representative human pose estimation datasets, MS COCO [16] and CrowdPose [14]. MS COCO contains 200K images and 250K person instances with 17 keypoint annotations per instance (we set the number of keypoint queries as $K = 17$ on MS COCO). CrowdPose has 20K images and 80K person instances with 14 keypoint annotations per instance ($K = 14$ on CrowdPose). CrowdPose is more challenging as it includes many crowd and occlusion scenes. We train Group Pose on COCO `train2017` set and evaluate it on COCO `val2017` set and `test-dev` set. On CrowdPose, we train our model on the `train` set and evaluate it on the `test` set.

Evaluation metric. The OKS-based average precision (AP) scores are reported as the main metric for both

Table 1: **Comparisons with state-of-the-art methods on MS COCO `val2017`.** We also provide the reference (Ref) for previous frameworks. The 'HM', 'BR', and 'KR' denote heatmap-based losses, human box regression losses, and keypoint regression losses. 'RLE' represents the residual log-likelihood estimation in Poseur [21]. † denotes the flipping test. ‡ removes the prediction uncertainty estimation in Poseur as a fair regression comparison. The **best results** are highlighted in **bold**.

		Method	Ref	Backbone	Loss	AP	AP_{50}	AP_{75}	AP_M	AP_L
Non-End-to-End	Top-Down	Mask R-CNN [8]	CVPR 17	ResNet-50	HM	65.5	87.2	71.1	61.3	73.4
		Mask R-CNN [8]	CVPR 17	ResNet-101	HM	66.1	87.4	72.0	61.5	74.4
		PRTR† [15]	CVPR 21	ResNet-50	KR	68.2	88.2	75.2	63.2	76.2
		Poseur‡ [21]	ECCV 22	ResNet-50	RLE	70.0	–	–	–	–
		Poseur [21]	ECCV 22	ResNet-50	RLE	74.2	89.8	81.3	71.1	80.1
	Bottom-Up	HrHRNet† [5]	CVPR 20	HRNet-w32	HM	67.1	86.2	73.0	61.5	76.1
		DEKR† [7]	CVPR 21	HRNet-w32	HM	68.0	86.7	74.5	62.1	77.7
		SWAHR† [20]	CVPR 21	HRNet-w32	HM	68.9	87.8	74.9	63.0	77.4
		LOGO-CAP† [37]	CVPR 22	HRNet-w32	HM	69.6	87.5	75.9	64.1	78.0
	One-Stage	DirectPose [31]	–	ResNet-50	KR	63.1	85.6	68.8	57.7	71.3
		CenterNet† [40]	–	Hourglass-104	KR+HM	64.0	–	–	–	–
		FCPose [22]	CVPR 21	ResNet-50	KR+HM	63.0	85.9	68.9	59.1	70.3
		InsPose [28]	ACM MM 21	ResNet-50	KR+HM	63.1	86.2	68.5	58.5	70.1
End-to-End	Previous Works	PETR [27]	CVPR 22	ResNet-50	HM+KR	68.8	87.5	76.3	62.7	77.7
		PETR [27]	CVPR 22	Swin-L	HM+KR	73.1	90.7	80.9	67.2	81.7
		QueryPose [36]	NeurIPS 22	ResNet-50	BR+RLE	68.7	88.6	74.4	63.8	76.5
		QueryPose [36]	NeurIPS 22	Swin-L	BR+RLE	73.3	91.3	79.5	68.5	81.2
		ED-Pose [38]	ICLR 23	ResNet-50	BR+KR	71.6	89.6	78.1	65.9	79.8
		ED-Pose [38]	ICLR 23	Swin-L	BR+KR	74.3	91.5	81.6	68.6	82.6
	Ours	GroupPose	–	ResNet-50	KR	72.0	89.4	79.1	66.8	79.7
		GroupPose	–	Swin-T	KR	73.6	90.4	80.5	68.7	81.2
		GroupPose	–	Swin-L	KR	**74.8**	**91.6**	**82.1**	**69.4**	**83.0**

datasets. For MS COCO, we adopt AP with different thresholds and different object sizes (medium and large), denoted as AP, AP_{50}, AP_{75}, AP_M, and AP_L, following the standard evaluation process[3]. On CrowdPose, to better evaluate the model performance in different crowded scenarios, we adopt AP with different thresholds and different crowding levels, denoted as AP, AP_{50}, and AP_{75}, as well as AP_E, AP_M and AP_H for images with easy, medium and hard crowding levels.

Implementation details. Our training and testing settings follow ED-Pose [38]. During training, we adopt the widely-used data augmentations in DETR frameworks [2, 41, 39, 38], including random flip, random crop, and random resize with the short sides in $[480, 800]$ and the long side less or equal to 1333. We use the AdamW optimizer[11, 19] with the weight decay 1×10^{-4} and train 60 epochs and 80 epochs on MS COCO [16] and CrowdPose [14], respectively. We adopt a total batch size of 16, and set the base learning rate as 1×10^{-4}. The base learning rate for the backbone is 1×10^{-5} following the DETR frameworks. The learning rates are decayed at the 50-th epoch and 70-th by a factor of 0.1 for MS COCO and CrowdPose, respectively. During testing, we resize the input images with their short sides being 800 and long sides less or equal to 1333.

4.2. Main Results

Our purpose is to build a simple baseline for end-to-end multi-person pose estimation. Thus, we mainly compare our Group Pose with previous end-to-end frameworks, including PETR [27], QueryPose [36], and ED-Pose [38]. Besides, to show the effectiveness of our method, we also add comparisons with non-end-to-end frameworks, such as top-down [8, 15], bottom-up [5, 7, 20, 37], and one-stage methods [31, 40, 22, 28].

Comparisons with end-to-end frameworks on COCO. Table 1 and Table 2 present the comparisons on COCO `val2017` set and `test-dev` set. Results show that Group Pose outperforms PETR [27], QueryPose [36], and ED-Pose [38] consistently.

On COCO `val2017`, Group Pose surpasses PETR and QueryPose by over a significant 3.0 AP with ResNet-50 [9] and the gaps remain 1.5+ AP with a strong backbone, Swin-Large [18]. When comparing with the recently proposed ED-Pose [38], which transfers pose estimation to a keypoint box detection problem and combines human box detection task, Group Pose can also exceed it with non-negligible margins. Moreover, unlike these methods that use complex

[3] https://cocodataset.org/#keypoints-eval

Table 2: **Comparisons with state-of-the-art methods on MS COCO `test-dev2017` dataset.** Notations are consistent with Table 1.

		Method	Ref	Backbone	Loss	AP	AP$_{50}$	AP$_{75}$	AP$_M$	AP$_L$
Non-End-to-End	Top-Down	Mask R-CNN [8]	CVPR 17	ResNet-50	HM	63.9	87.7	69.9	59.7	71.5
		Mask R-CNN [8]	CVPR 17	ResNet-101	HM	64.3	88.2	70.6	60.1	71.9
		PRTR† [15]	CVPR 21	ResNet-101	KR	68.8	89.9	76.9	64.7	75.8
		PRTR† [15]	CVPR 21	HRNet-w32	KR	71.7	90.6	79.6	67.6	78.4
	Bottom-Up	HrHRNet† [5]	CVPR 20	HRNet-w32	HM	66.4	87.5	72.8	61.2	74.2
		DEKR† [7]	CVPR 21	HRNet-w32	HM	67.3	87.9	74.1	61.5	76.1
		SWAHR† [20]	CVPR 21	HRNet-w32	HM	67.9	88.9	74.5	62.4	75.5
		LOGO-CAP† [37]	CVPR 22	HRNet-w32	HM	68.2	88.7	74.9	62.8	76.0
	One-Stage	DirectPose [31]	–	ResNet-50	KR	62.2	86.4	68.2	56.7	69.8
		CenterNet† [40]	–	Hourglass-104	KR+HM	63.0	86.8	69.6	58.9	70.4
		FCPose [22]	CVPR 21	ResNet-50	KR+HM	64.3	87.3	71.0	61.6	70.5
		InsPose [28]	ACM MM 21	ResNet-50	KR+HM	65.4	88.9	71.7	60.2	72.7
End-to-End	Previous Works	PETR [27]	CVPR 22	ResNet-50	HM+KR	67.6	89.8	75.3	61.6	76.0
		PETR [27]	CVPR 22	Swin-L	HM+KR	70.5	91.5	78.7	65.2	78.0
		QueryPose [36]	NeurIPS 22	Swin-L	BR+RLE	72.2	92.0	78.8	67.3	79.4
		ED-Pose [38]	ICLR 23	ResNet-50	BR+KR	69.8	90.2	77.2	64.3	77.4
		ED-Pose [38]	ICLR 23	Swin-L	BR+KR	72.7	92.3	80.9	67.6	80.0
	Ours	GroupPose	–	ResNet-50	KR	70.2	90.5	77.8	64.7	78.0
		GroupPose	–	Swin-T	KR	72.1	91.4	79.9	66.7	79.5
		GroupPose	–	Swin-L	KR	**72.8**	**92.5**	**81.0**	**67.7**	**80.3**

decoders and add extra supervisions, *e.g.*, extra heatmap or box supervisions, our Group Pose only use a simple decoder and are trained only with keypoint regression targets. The above evidences indicate that complex design for end-to-end multi-person pose estimation may not be necessary.

On COCO `test-dev`, our Group Pose achieves 70.2 AP, 72.1 AP, and 72.8 AP with ResNet-50 [9], Swin-Tiny [18], and Swin-Large [18] as the backbone. Compared with other end-to-end frameworks, similar trends are observed with the ones on COCO `val2017`.

Comparisons with end-to-end frameworks on CrowdPose. To further demonstrate the effectiveness of Group Pose, we provide comparisons with previous end-to-end frameworks on the challenging CrowdPose dataset [14]. Table 3 reports the results of PETR [27], QueryPose [36], ED-Pose [38], and our Group Pose with the same backbone Swin-Large [18]. Overall, Group Pose achieve 74.1 AP, performing the best over all methods.

Moreover, it is interesting when we compare the AP scores with easy, medium, and hard crowding levels of different models. PETR [27] performs worse with the easy crowding level while giving the second best results with the hard level. ED-Pose [38] performs worst with the hard crowding level. We conjecture that this phenomenon is caused by the differences in their decoders, *e.g.*, human instances crowded with each other have similar boxes, which challenges the human keypoint decoder. Our Group Pose, which uses K keypoint queries and one instance query for each human instance and considers different interactions among queries, can achieve reasonable good results with easy, medium, and hard crowding levels.

Table 3: **Comparisons with state-of-the-art methods on CrowdPose `test` dataset.** Swin-L is adopted as the backbone. Other notations are consistent with Tabel 1.

Method	Loss	AP	AP$_{50}$	AP$_{75}$	AP$_E$	AP$_M$	AP$_H$
PETR [27]	HM+KR	71.6	90.4	78.3	77.3	72.0	65.8
QueryPose [36]	BR+RLE	72.7	**91.7**	78.1	79.5	73.4	65.4
ED-Pose [38]	BR+KR	73.1	90.5	79.8	80.5	73.8	63.8
GroupPose	KR	**74.1**	91.3	**80.4**	**80.8**	**74.7**	**66.4**

Comparisons with various non-end-to-end frameworks on COCO. We also compare Group Pose with representative non-end-to-end frameworks in Table 1 and Table 2. Group Pose can easily outperform previous bottom-up methods [5, 7, 20, 37] and one-stage methods [31, 40, 22, 28]. For example, Group Pose surpasses the recent proposed LOGO-CAP [37] with flipping test by over 2.0 AP on both COCO `val2017` and `test-dev`, even with a smaller backbone (ResNet-50 [9]) than it (HRNet-w32 [29]). Group Pose is more concise and precise than those bottom-up and one-stage methods. Surprisingly, Group Pose can also beat previous top-down methods like PRTR [15] and Poseur [21]. Given the simple design and the end-to-end property, our Group Pose can serve as a simple baseline for pursuing higher performance on multi-person pose estimation.

Table 4: **Ablation experiments for Group Pose.** Evaluated on MS COCO `val2017`. Default settings are marked in gray .

(a) **Query designs for human instances.** Both instance (inst) and keypoint (kpt) queries are essential in Group Pose, especially the keypoint ones.

query types	AP	AP_M	AP_L
inst & kpt	**72.0**	**66.8**	**79.7**
only kpt	71.2	66.0	79.1
only inst	64.5	61.1	69.9

(b) **Benefits of the instance query.** On the model only with kpt queries, the inst query is first added but not for classification (cls). Then we use the inst query for the cls task.

query types	cls task	AP	AP_M	AP_L
only kpt	avg kpt	71.2	66.0	79.1
inst & kpt	avg kpt	71.7	66.8	79.3
inst & kpt	inst	**72.0**	**66.8**	**79.7**

(c) **Number of instances.** When the number is smaller than 100, increasing instance numbers provide gains. 100 is set by default for the number of instances.

#instance	AP	AP_M	AP_L
50	71.4	66.4	79.1
100	**72.0**	66.8	**79.7**
200	**72.0**	**66.9**	**79.7**

(d) **Self-attention implementations.** In the self-attention module, removing the across-instance interactions for queries with different types with attention (attn) mask or group self-attentions eases the optimization and improves the performance.

self-attention implementations	w/ attn mask	AP	AP_M	AP_L
standard self-attention	×	69.4	64.0	77.5
standard self-attention	✓	70.7	65.5	78.4
group self-attentions	×	**72.0**	**66.8**	**79.7**

(e) **Group self-attentions.** Both types of self-attentions are important in group self-attentions. We also conduct an experiment by removing both within-instance and across-instance self-attentions. The performance further drops to 65.3 AP.

group self-attentions	AP	AP_M	AP_L
wthin-instance & across-instance self-attentions	**72.0**	**66.8**	**79.7**
within-instance self-attentions	66.3	63.8	70.8
across-instance self-attentions	67.4	63.0	74.4

4.3. Ablation Study

We run a number of ablation experiments to verify the effectiveness of key elements in our Group Pose. We adopt ResNet-50 [9] as the backbone. Unless specified, we report the results on COCO `val2017` with 60 epochs training.

Ablation: query designs for human instance. Table 4a gives the results of representing a human instance with different query designs. Both instance query and keypoint queries are essential in Group Pose. As the multi-person pose estimation is to predict human poses given an image, it is reasonable that keypoint queries themself can achieve good results, while removing keypoint queries gives a significant performance drop (from 72.0 AP to 64.5 AP).

For the instance query, its benefits come from two aspects: (i) gather information within human instance and help model training and (ii) decouple the classification task and keypoint positions regression task. Table 4b shows the improvements brought by these two benefits.

Ablation: self-attention implementations. Table 4d provides the comparisons between different self-attentions in the decoder layers, including (i) standard self-attention over all the $N \times (K+1)$ queries (Figure 3 (a)), (ii) standard self-attention with an attention mask (Figure 3 (b)), which masks out the across-instance interactions for queries with different types, and (iii) our group self-attentions (Figure 3 (c)). Results validate that the across-instance interactions for queries with different types are not directly helpful for the multi-person pose estimation task. Removing this type of interactions eases the model optimization, helps model converge faster, thus improves the performances. The comparisons on convergence curves are given in Figure 4. Besides, we find that it is better to separately perform different types of interactions, the within-instance and same-type across-instance interactions, with different parameters. Table 4d shows that our sequential implementation group self-attentions bring a 1.3 AP gain over the parallel implementation of standard self-attention with an attention mask.

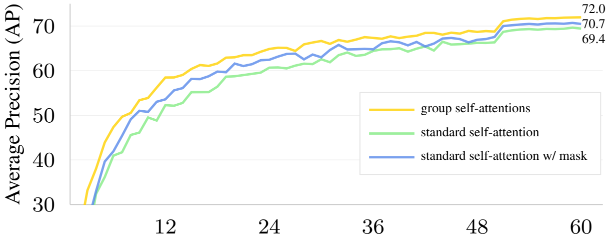


Figure 4: **Faster and better convergence.** The x-axis corresponds to #epoch, and the y-axis corresponds to AP score. One can see that group self-attentions converge faster and better than other self-attention implementations.

Ablation: group self-attentions. We ablate the effects of N within-instance self-attentions over K keypoint queries and one instance query and $(K+1)$ across-instance self-attentions over N queries of the same type in Table 4e. Large performance gaps (over -5.0 AP) are observed when we remove either of these two group self-attentions. Both the interactions are important for multi-person pose estimation. The within-instance self-attentions explore kinematic relations and gather information for scoring pose predictions, which help the model predict precise keypoint positions for human poses. The same-type across-instance self-attentions collect information from the same-type of keypoint queries or instance queries, removing duplicate predictions for poses among human instances. As shown in Figure 5, model without the same-type across-instance self-attentions produces more duplicated pose predictions for the same human instance than Group Pose.

Ablation: number of instances. Table 4c gives how predefined number of human instances affect the results. We find 100 human instances are enough and can achieve comparable results with 200 human instances. We set the num-

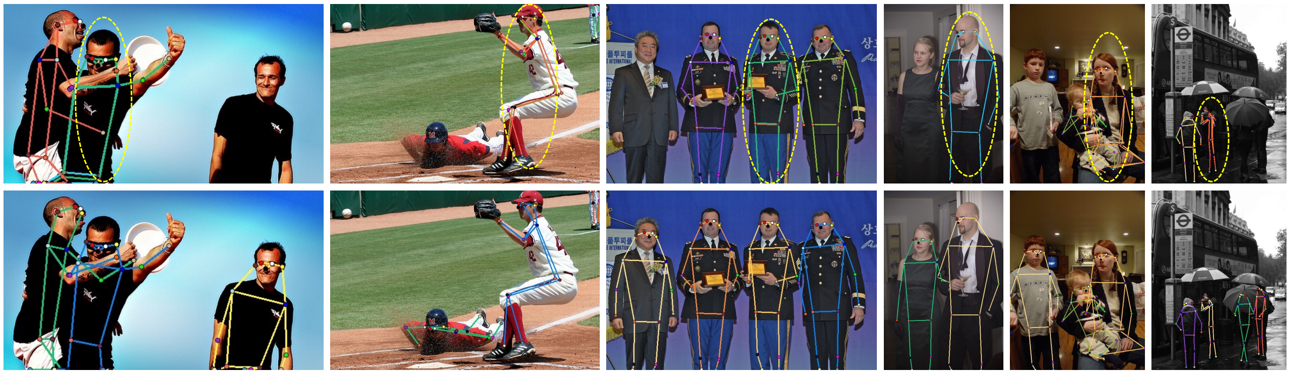


Figure 5: **Comparison of removing duplicated predictions.** We visualize the predicted poses of Group Pose (second row), and Group Pose without same-type across-instance self-attentions (first row) according to the number of objects. Group Pose produces less duplicated pose predictions. The yellow dashed ellipse indicates the duplicated predicted human instance, leading to a mismatch of other instances. Best view in zoom in.

Table 5: **Analysis on model convergence.** Group Pose without human detection already can outperform ED-Pose [38]. Besides, Group Pose can also benefit from better human instance initialization with human detection. We report AP on COCO `val2017` dataset. 'Det Dec' = human detection decoder. The results of ED-Pose are from the original paper [38].

Method	w/ Det Dec	12e	24e	36e	48e	60e
ED-Pose [38]	✓	60.5	67.5	69.7	70.8	71.6
GroupPose	×	**61.0**	**67.6**	**70.1**	**71.4**	**72.0**
GroupPose	✓	**61.4**	**68.1**	**70.3**	**71.6**	**72.2**

ber of human instances as 100 by default. Thus, Group Pose has $100 \times (17 + 1)$ queries on COCO [16] and contains $100 \times (14 + 1)$ queries on CrowdPose [14].

4.4. More Analysis

We provide more analysis about our Group Pose in this section, including the analysis on model convergence and the analysis on model inference speed, detailed next.

Analysis on model convergence. A common wisdom about training models is to provide a good initialization and then refine the model based on it. ED-Pose [38] splits the multi-person pose estimation task into two sub-processes, which first detect human instances with a human detection decoder and then use a human-to-keypoint detection decoder for predicting human poses. The human detection decoder gives good initialization for human instances, which can help ED-Pose learn and converge faster. Although our Group Pose adopts a different design for transformer decoder, it can also be built upon a human detection decoder and enjoys the benefits brought by better initialization of human instances. Based on this observation, we build a variant of our Group Pose, whose transformer decoder consists of 2 human detection decoder layers and 4 simple transformer decoder layers for multi-person pose estimation, following ED-Pose [38].

Table 6: **Analysis on model inference speed.** Frames per second (FPS) and inference time (Time) are measured with ResNet-50 and different image resolutions on one NVIDIA A100 GPU.

Method	Input Resolution	FPS ↑	Time [ms] ↓
PETR [27]	480 × 800	20.0	50
	800 × 1333	12.1	83
QueryPose [36]	480 × 800	19.0	56
	800 × 1333	13.4	75
ED-Pose [38]	480 × 800	42.4	24
	800 × 1333	24.7	40
GroupPose	480 × 800	**68.6**	**15**
	800 × 1333	**31.3**	**32**

Table 5 shows the AP scores of three models, ED-Pose, Group Pose, and Group Pose with human detection decoder, under different training schedules. Group Pose already shows superior results than ED-Pose with 12 epochs, 24 epochs, 36 epochs, 48 epochs, and 60 epochs training, even without the help of human detection decoder. Moreover, the comparison between Group Pose and Group Pose with human detection decoder verifies that good initialization for human instances leads to faster convergence and better results.

Analysis on model inference speed. Table 6 provides the comparisons of inference time and FPS among end-to-end frameworks, including PETR [27], QueryPose [36], ED-Pose [38], and our Group Pose. Simple designs usually show efficiency. Results show that our simple transformer decoder can be faster than complex decoders in previous end-to-end frameworks. Even with an image in a large size 800 × 1333, our Group Pose can also achieve real-time speed (above 30 FPS) on a single A100 GPU.

5. Conclusion

In this paper, we present a simple baseline, Group Pose, for end-to-end multi-person pose estimation. With simple designs for queries and decoder self-attentions, the approach outperforms previous end-to-end frameworks while being faster. Besides, the transformer decoder in Group Pose is also flexible, which can be built solely or upon human detection decoders. We hope Group Pose can provide insights for exploring concise and effective end-to-end multi-person pose estimation frameworks.

Acknowledgements. This work was supported in part by the National Key R&D Program of China (No.2021ZD0112100), National NSF of China (No.U1936212, No.62120106009, No.62261160653).

References

[1] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017. 1, 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 1, 2, 3, 4, 5

[3] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022. 2

[4] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018. 1, 2

[5] 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, 2, 5, 6

[6] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2334–2343, 2017. 1, 2

[7] 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. 5, 6

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2961–2969, 2017. 2, 5, 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 3, 5, 6, 7

[10] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4507–4515, 2017. 1, 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] 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

[13] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 2

[14] 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. 2, 4, 5, 6, 8

[15] 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. 5, 6

[16] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 2, 4, 5, 8

[17] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 2

[18] 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*, pages 10012–10022, 2021. 2, 3, 5, 6

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[20] 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. 5, 6

[21] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 72–88, 2022. 5, 6

[22] 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, 5, 6

[23] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. 2, 4

[24] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2

[25] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 483–499. Springer, 2016. 1, 2

[26] 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. 1, 2

[27] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11069–11078, 2022. 1, 2, 3, 5, 6, 8

[28] 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. 5, 6

[29] 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, 6

[30] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021. 2

[31] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *arXiv preprint arXiv:1911.07451*, 2019. 1, 2, 5, 6

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[33] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 1, 2

[34] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 527–544. Springer, 2020. 1, 2

[35] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision*, pages 466–481, 2018. 1, 2

[36] Yabo Xiao, Kai Su, Xiaojuan Wang, Dongdong Yu, Lei Jin, Mingshu He, and Zehuan Yuan. Querypose: Sparse multi-person pose regression via spatial-aware part-level query. In *Advances in Neural Information Processing Systems*, pages 1–14, 2022. 2, 3, 5, 6, 8

[37] Nan Xue, Tianfu Wu, Gui-Song Xia, and Liangpei Zhang. Learning local-global contextual adaptation for multi-person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13065–13074, 2022. 5, 6

[38] Jie Yang, Ailing Zeng, Shilong Liu, Feng Li, Ruimao Zhang, and Lei Zhang. Explicit box detection unifies end-to-end multi-person pose estimation. In *International Conference on Learning Representations*, pages 1–17, 2023. 1, 2, 3, 5, 6, 8

[39] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, pages 1–18, 2022. 1, 2, 5

[40] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 5, 6

[41] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, pages 1–16, 2021. 1, 2, 3, 4, 5