Object-Centric Multi-Task Learning for Human Instances

Hyeongseok Son
hs1.son@samsung.com
Sangil Jung
sang-il.jung@samsung.com
Solae Lee
solae913.lee@samsung.com
Seongeun Kim
se91.kim@samsung.com
Seung-In Park
si14.park@samsung.com
ByungIn Yoo
byungin.yoo@samsung.com

Samsung Advanced Institute of Technology (SAIT)
Suwon, Republic of Korea

Abstract

Human is one of the most essential classes in visual recognition tasks such as detection, segmentation, and pose estimation. Despite considerable efforts in addressing these tasks individually, their integration within a multi-task learning framework has been relatively unexplored. In this paper, we explore a compact multi-task network architecture that maximally shares the parameters of the multiple tasks via object-centric learning. To this end, we introduce a novel human-centric query (HCQ) that effectively encodes human instance information, including explicit structural information such as keypoints. Besides, we utilize HCQ in prediction heads of the target tasks directly and also interweave HCQ with the deformable attention in Transformer decoders to exploit a well-learned object-centric representation. Experimental results show that the proposed multi-task network achieves comparable accuracy to state-of-the-art task-specific models in human detection, segmentation, and pose estimation tasks, while it consumes less computational costs. The project page is available at this https URL

1 Introduction

Core tasks in visual recognition, such as detection, segmentation, and pose estimation, play a crucial role in diverse applications, including video surveillance and human-computer interaction. Adopting a multi-task learning strategy with a unified architecture, rather than training individual models for each task, offers cost-efficiency and promotes inter-task synergy. In this paper, our focus lies in developing an effective unified architecture tailored specifically for human-related multi-task learning.

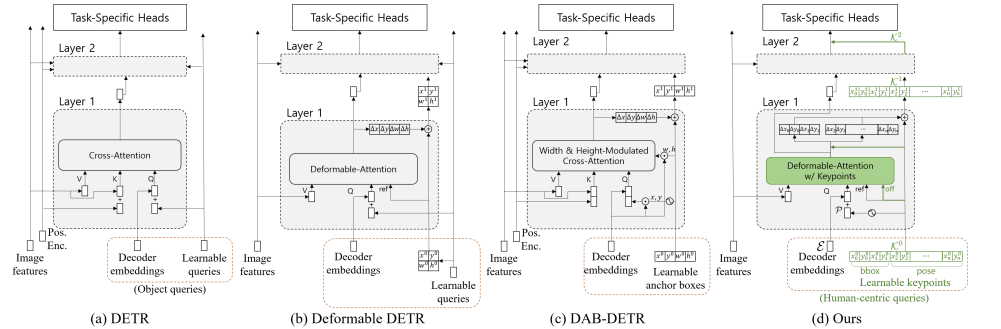

Figure 1: Query design comparison. We visualize the cross-attention part in Transformer decoders, highlighting the main differences in green. Compared to DETR [2], Deformable DETR [29] and DAB-DETR [15] embed explicit positional information of an object to a learnable query. Our human-centric query additionally incorporates object structural information as keypoints, allowing it to effectively carry diverse information for multiple tasks. In contrast to other methods, these learnable keypoints are directly fed into task-specific heads, providing high-level positional and structural information and enabling each head to jointly consider it. Moreover, learnable keypoints serve as sampling locations in deformable attentions, accelerating the learning process and boosting multi-task performance.

Recent advances in object-centric learning [2, 17] use the Transformer architecture to encode per-instance representations by mapping Transformer queries to image object instances. Due to its per-instance representation capabilities, object-centric learning can be well-suited for human instance-level recognition tasks. Although some studies apply object-centric learning to multi-tasking of detection and segmentation [11], few have extended this concept to human recognition problems. Moreover, human recognition tasks, unlike general object recognition, necessitate pose estimation, which conveys the structural information of human instances. A naïve application of object-centric learning to multi-task learning could cause performance degradation, as different types [25] of information are implicitly encoded in a mixed manner. This prevents each query from possessing task-specific information, which is essential for effective multi-task learning in human recognition problems.

To tackle this problem, we propose a novel human-centric query (HCQ) design to extract more representative information of human instances. We segregate the structural information from the *decoder embedding* and represent it in explicit forms called *learnable keypoints* (Fig. 1d). The learnable keypoints contain the bounding box and the body joints of a human while the decoder embedding encodes the general representation of an instance. This separation enables different pieces of information of a human instance to be encoded in a decoupled way, allowing multiple target tasks to effectively utilize useful information from the query for their own purposes.

Thanks to the decoupled design of HCQ, only the light-weight prediction heads are needed because the information is already disentangled enough for each task. In addition, utilizing the decoupled learnable keypoints in the prediction heads further improves the performance because it gives extra pre-computed and high-quality structural information that the other query does not possess. For example, human pose can help segmentation [27] and *vice versa*. While some previous methods (e.g., DAB-DETR [15]) explicitly represent coarse information like bounding boxes, this representation may not provide enough structural information to assist with other tasks, such as segmentation. Hence, most previous methods feed only decoder embeddings containing general information into the prediction heads to

perform target tasks even though they have explicit box coordinates.

Furthermore, our learnable keypoints can be interweaved with deformable cross-attention seamlessly. Deformable attention [29] predicts sample locations directly from decoder embeddings rather than computing similarity between the query and image features. Our learnable keypoints can be directly applied to deformable cross-attention as sampling locations. In other words, pre-computed learnable keypoints carrying bounding box and pose information can be reused as attention, reducing redundant computations. We discovered several benefits: attention learning supervised by poses accelerates the training process, and structural information in human poses provides additional clues for occluded regions.

We evaluate our method on the COCO dataset [13] through various experiments. To our best knowledge, ours is the first Transformer-based unified architecture for human-related multi-tasks, which are considered less correlated in task taxonomy (e.g., Semantic Segmentation vs. 2D Keypoints) [25]. Our approach achieves comparable accuracy to state-of-the-art task-specific models while maintaining a compact design.

2 Related Work

Object-centric query design Locatello et al. [17] introduce the concept of object-centric representation learning to separate image information based on individual object instances. They use a set of object representations, called slots, and map them to image object instances using slot attention. DETR [2], a concurrent work, applies the Transformer architecture for object detection using object-centric representations. They employ a fixed-size set of learnable object queries to infer object relations and image features (Fig. 1(a)). Many follow-up approaches [10, 15, 19, 23, 26, 29] have tackled the object detection task based on DETR. We focus on representative methods concerning object query designs. Deformable DETR [29] uses 2D reference points from learned linear projection (Fig. 1(b)), enabling faster convergence and improved object detection. DAB-DETR [15] replaces queries with dynamic anchor boxes containing position and size information (Fig. 1(c)), allowing decoders to concentrate on regions of interest. Distinct from the previous works, our learnable query explicitly and compactly carries the structural information of an object as the form of keypoints (Fig. 1(d)). Furthermore, we present effective ways to exploit the high-level information learned in the query for performing deformable attention and task-specific heads.

Object-centric multi-task learning Object-centric representation has been applied to general object multi-task learning for detection and segmentation. DETR [2], an object detector utilizing object queries to communicate with image features, can be easily extended to segmentation tasks by attaching mask heads. However, its segmentation performance is inferior to task-specific models. From a segmentation perspective, semantic, instance, and panoptic [9] segmentation can be considered distinct tasks, with some research efforts [4, 5, 28] aiming to address all of them within a unified architecture. Mask2Former [4, 5] attends only the masked regions in cross-attention for fast convergence and K-net [28] introduces learnable kernels with update strategy where each kernel is in charge of each mask. Recently, Mask-DINO [11] investigates joint performance improvement for detection and segmentation tasks using anchor box-guided cross-attention [26] and a denoising training scheme [10]. However, none of these approaches have explored object-centric query design for multi-task learning that includes more heterogeneous tasks, such as pose estimation.

Multi-task learning for human instances Multi-person instance segmentation and pose estimation [1, 21, 27] are essential for human-related visual tasks. PersonLab [21] adopts

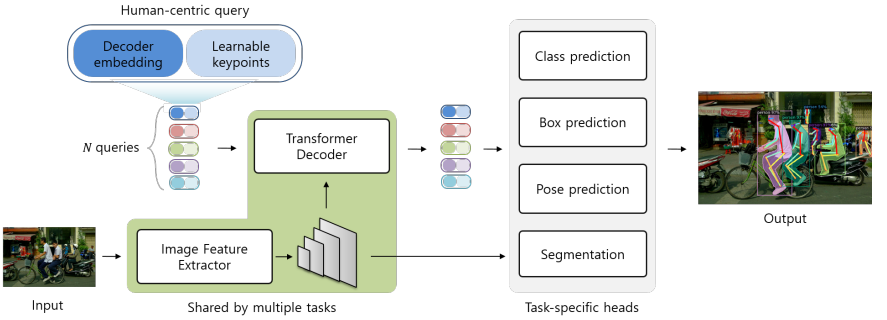

Figure 2: Our unified network architecture for various instance-level vision tasks; human detection, segmentation and pose estimation.

a bottom-up approach, first extracting multiple feature maps and then associating them to obtain instance-wise segmentation and pose. Pose2seg [27] uses human poses instead of the bounding boxes to normalize interest regions for better alignment and performs the target task for each proposal. PosePlusSeg [1] uses a shared backbone to get intermediate maps for each task and each task pipeline combines those maps to get the results. In contrast to the aforementioned convolution-based methods, we employ a Transformer decoder architecture with human-centric queries to perform these tasks simultaneously.

3 Method

Overall architecture The overall architecture (Fig. 2) has three components: an image feature extractor, a Transformer decoder, and task-specific heads. The image feature extractor takes an image as input and produces multi-resolution features. These features are fed into the Transformer decoder for attention. The decoder processes human-centric queries with various human instance information, which are then fed into lightweight task-specific heads for final prediction. This design results in a lightweight multi-task model.

3.1 Human-centric query design

Since a single query vector for training multi-tasks represents tangled information and usually causes performance degradation, we design a novel human-centric query to decouple the structural information from the tangled embedding vector. The proposed human-centric query consists of two distinct parts, *decoder embeddings* and *learnable keypoints* (Fig. 3). The decoder embedding $\mathcal{E}_q^l \in \mathbb{R}^{1\times D}$ is the representation vector for each human instance where q and l are indices for query and layer, respectively, and D is the hidden dimension for each query. The learnable keypoints $\mathcal{K}_q^l \in \mathbb{R}^{1\times 2(n+1)}$ are the decoupled positional and structural information of a human instance and defined as

$$\mathcal{K}_q^l = \left[x_{q,0}^l, y_{q,0}^l, \cdots, x_{q,n}^l, y_{q,n}^l\right] = \left[\mathbf{p}_{q,0}^l, \cdots, \mathbf{p}_{q,n}^l\right],$$

where $\mathbf{p}_{q,i}^l := (x_{q,i}^l, y_{q,i}^l)$ is a 2D coordinate of the ith keypoint. Emphasizing that the formulation is identical for all queries and layers, we omit the indices q and l without loss of generalities.

The learnable keypoints $\mathcal{K}$ contains two different types of information; a *bbox* part $(\mathbf{p}_0, \mathbf{p}_1)$ and a *pose* part $(\mathbf{p}_2, \ldots, \mathbf{p}_n)$. Keypoints $\mathbf{p}_0$ and $\mathbf{p}_1$ in the *bbox* part give xy-coordinates of left-top and right-bottom of the bounding box, respectively. These two diagonal points are

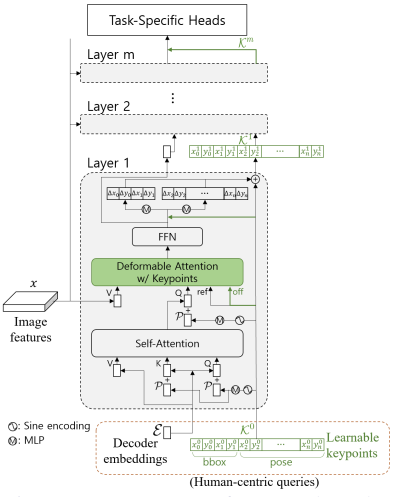

Figure 3: Our transformer decoder architecture.

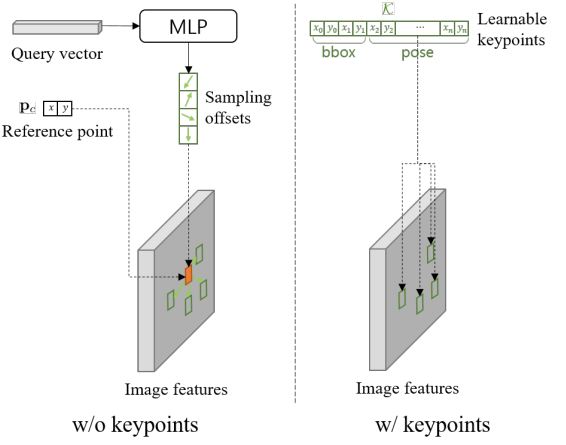

Figure 4: Sampling locations of deformable attention w/o and w/ keypoints.

sufficient to represent a given bounding box and we define the center and side lengths of the bounding box using these two points:

$$\mathbf{p}_c = \frac{\mathbf{p}_0 + \mathbf{p}_1}{2} \quad \text{and} \quad \mathbf{d} = \mathbf{p}_1 - \mathbf{p}_0 \,. \tag{1}$$

Unlike the *bbox* part, keypoints in the *pose* part are regarded as coordinates in a canonical space defined by the bounding box, i.e., the set $\mathcal{J}$ of xy-coordinates of joints in the human pose are computed as

$$\mathcal{J} = \{\mathbf{p}_0 + \mathbf{p}_i T \mid i = 2, \ldots, n\} \,, \tag{2}$$

where $T := \mathrm{diag}(\mathbf{d})$ is a dilation operator. We represent the joint cooridnates of pose in the canonical space which is normalized by the box. It reduces the pose variance according to the size of the human so that it lessens the burden of the network. The effectiveness of the canonical space can be found in the supplementary material.

Inspired by the DAB-DETR [15], the corresponding structural embedding $\mathcal{P} \in \mathbb{R}^{1\times D}$ is obtained by successive operations over $\mathcal{K}$. First, a sine encoding $\sigma\colon \mathbb{R} \to \mathbb{R}^{1\times D'}$ maps each element of $\mathcal{K}$ to a vector and then, a multi-layer perceptron MLP is applied. In other words,

$$\mathcal{P} = \mathrm{MLP}\left(\sigma(\mathcal{K})\right) = \mathrm{MLP}\left(Cat\left(\sigma(x_0), \sigma(y_0), \ldots, \sigma(x_n), \sigma(y_n)\right)\right) \,, \tag{3}$$

where *Cat* is a concatenation operator along with the last dimension. Here, MLP is a three-layer perceptron:

$$\mathrm{MLP}(\mathbf{x}) = \mathrm{ReLU}\left(\mathrm{ReLU}\left(\mathbf{x}W_1 + \mathbf{b}_1\right)W_2 + \mathbf{b}_2\right)W_3 + \mathbf{b}_3 \,,$$

where $W_1 \in \mathbb{R}^{2(n+1)D'\times D}$, $W_2, W_3 \in \mathbb{R}^{D\times D}$, and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 \in \mathbb{R}^{1\times D}$ are learnable weights and biases. One notable point is that our learnable keypoints carries salient coordinates for not only bounding box but also joints of human pose. This allows each human-centric query to become an expert on position and structure of the corresponding human object.

3.2 Query utilization in task-specific heads

Previous works do not use a learnable query as an input of task-specific head networks, even though the query contains high-level information pre-computed from previous layers, such as

the coordinates of a bounding box (and also the joints of a pose in our approach). We simply feed the concatenation of the coordinates along with decoder embeddings into task-specific heads as the form of conditional information. By doing so, each task head can consider useful information in other tasks (bbox and pose estimation) for improving its accuracy.

In Fig. 3, regarding object detection and pose estimation tasks, as learnable keypoints themselves are the results of the tasks, their task heads perform in every layer in the transformer decoder. Other task heads perform only once, after the transformer decoder. For all these task heads, we provide the information of learnable keypoints.

Pose estimation head While other task-specific heads employ conventional structures used in DETR [2] and MaskFormer [4], our pose estimation head network has a simple architecture, distinguished from previous pose estimation methods. Our head network receives object queries and produces the coordinates of pose joints directly. Off-the-shelf pose estimation methodologies [3, 22, 27] used auxiliary expedients such as bounding box cropping & resizing or heat map extraction. While existing methods offer performance advantages, they have limitations in reducing computational cost. Our approach, benefiting from object-centric queries, employs a vector-to-vector head network that directly regresses pose joints from the human-centric query. This results in lower computational cost compared to conventional task heads that process cropped images and produce heat maps (e.g., 304-to-34 in our head vs. $256 \times 256 \times 3$-to-$56 \times 56 \times 17$ in [3] for each instance).

3.3 Query utilization in deformable attentions

Our learnable query carries keypoint coordinates with high accuracy for a bounding box and a pose, and keypoints refer to salient points by definition. By reusing such information as sampling points in an attention module, we can reduce redundant computations for attention calculation. In this regard, we directly use our learnable keypoints as sampling locations in a deformable attention layer (Fig. 4 right).

Deformable attention with keypoints Let m be an attention head index and let $Q \in \mathbb{R}^{1\times D}$ be the query vector which is sum of $\mathcal{P}$ and an output of self-attention module. The set of proposed sampling locations $\mathcal{S}_m$ is a union of two subsets; the set $\mathcal{D}_m$ of Deformable DETR sampling locations and the set $\mathcal{J}_m$ of joint coordinates for *pose* (Fig. 4). The sampling locations in $\mathcal{D}_m$ are decided by sampling offsets $\Delta\mathbf{p}_m$ which are generated by MLP over the query vertor Q:

$$\mathcal{D}_m = \{\mathbf{p}_c + \Delta\mathbf{p}_m \mid \forall \text{generated } \Delta\mathbf{p}_m\}\,. \tag{4}$$

Note that the center of the bounding box $\mathbf{p}_c$ in Eq. (1) plays role of the reference point. The joint coordinates for each head $\mathcal{J}_m$ are equally split from $\mathcal{J}$ in Eq. (2). For the number N_h of attention heads and the number N_p of all sampling locations in each head, let $x \in \mathbb{R}^{H\times W\times D}$ be an image feature map, and let $W_m \in \mathbb{R}^{D\times D/N_h}$ be a learnable weight. The value $V_m \in \mathbb{R}^{N_p\times D/N_h}$ is defined as a stack of sampled features at xW_m:

$$V_m = Cat\left(\left(xW_m(\mathbf{p})\right)^T\right)^T, \tag{5}$$

where the concatenation is applied across for all $\mathbf{p} \in \mathcal{S}_m$. From this, the output of the deformable attention with keypoints can be represented as

$$\text{DeformAttKey}(Q, V, \mathcal{P}) = Cat\left(A_1V_1, \ldots, A_{N_h}V_{N_h}\right)W\,, \tag{6}$$

where $A_m \in \mathbb{R}^{1\times N_p}$ is an attention coefficient obtained by linear operator over Q and $W \in \mathbb{R}^{D\times D}$ is a learnable weight. Eq. (6) can naturally be used to the case of multi-scale attention module. The mathematical derivation can be found in the supplementary material.

Model	Components			Accuracy (mAP)		
	Learnable keypoints	Query util. in T.H.	Query util. in D.A.	Det.	Pose.	Seg.
BaseNet-DS				56.4	✗	51.1
BaseNet-DPS				53.5	55.7	49.3
HCQNet-α	✓			55.9	33.3	50.9
HCQNet-β	✓	✓		56.8	60.2	51.5
HCQNet	✓	✓	✓	56.1	64.4	51.7

Table 1: Ablation study on query design. Query util. in T.H. and Query util. in D.A. mean query utilization in task-specific heads and in deformable attentions, respectively.

To address potential regions not covered by keypoints, we also use original sampling locations alongside, predicted by an MLP as in Deformable DETR [29]. Specifically, for 32 sampling locations per object query, we obtain 16 locations from the keypoints and the remaining 16 locations following Eq. (4) (Fig. 4 left).

4 Experimental Results

Settings We use MS COCO 2017 dataset [13] for network training and validation as all labels for human pose estimation, detection, and segmentation tasks are provided. There might be a task-specific augmentation technique to obtain the best performance for each task. However, we applied one common data augmentation technique in [5] because the results of three tasks are obtained from a single image and we need to train all tasks at the same time. We use the AdamW [18] optimizer with the initial learning rate of 10^{-4} and the batch size of 16. We train each model for 368,750 iterations, and apply a learning rate decay at the two iterations of 327,778 and 355,092 with the decay value of 0.1. More details of the experimental setting including loss functions can be found in the supplementary material.

4.1 Component analysis

Ablation study Our baseline network is a careful combination of Mask2Former [5], Deformable DETR [29], and DAB-DETR [15]. In summary, we replace their cross-attention layers in Mask2Former with deformable attention layers. To predict reference points for deformable attention, we replace learnable query with the *bbox* part of learnable keypoints.

Utilizing the described architecture, we define two baseline models, BaseNet-DS and BaseNet-DPS (Table 1). BaseNet-DS handles detection and segmentation tasks only. Notably, due to our careful design, it achieves detection and segmentation performance comparable to state-of-the-art task-specific models (Mask2Former and DAB-DETR). In addition to BaseNet-DS, we consider multi-tasking with an additional human pose estimation task. We add a head network and an RLE loss function [12] for human pose estimation to the baseline model, which we refer to as the baseline multi-tasking model (BaseNet-DPS).

In this ablation study (Table 1), from BaseNet-DPS, we add our components of 1) learnable keypoints including also the joint coordinates of a pose, 2) query utilization in task-specific heads (Query util. in T.H.), and 3) query utilization in deformable attentions (Query util. in D.A.) one by one, and analyze their performance.

Simply adding a heterogeneous task causes the significant degradation of overall performance, showing a difficulty of learning shared representation for multi-tasking (BaseNet-DS vs. BaseNet-DPS). Using our query design, detection and segmentation accuracy increases (BaseNet-DPS vs. HCQNet-α), showing that explicit separation of mixed information im-

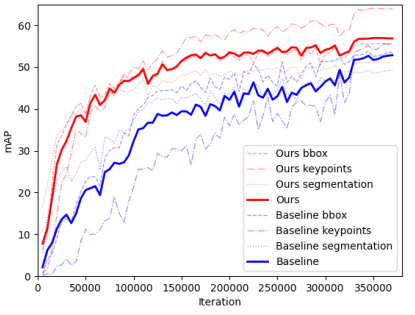

Figure 5: Training accuracy graph (Blue: BaseNet-DPS, Red: HCQNet).

Figure 6: Qualitative results on the COCO 2017 Person dataset [13].

proves accuracy. However, pose estimation accuracy drops, as decoder embeddings lack sufficient pose information.

Providing learnable keypoints to task-specific heads improves accuracy for all tasks, as positional and structural information complement each other (HCQNet-α vs. HCQNet-β). This is more effective in pose estimation than detection, despite similar prediction processes. In detection, the simpler box representation allows the prediction head to estimate proper displacement without the previously predicted bounding box, unlike in pose estimation.

Using learnable keypoints in deformable attentions can yield additional performance gains (HCQNet-β vs. HCQNet). In our experiment, this significantly improves pose estimation accuracy, has a smaller impact on segmentation, and slightly reduces detection accuracy. Overall, our components (HCQNet) significantly enhance performance for all target tasks compared to the baseline multi-tasking model (BaseNet-DPS).

Training speed In addition to the overall improvement in accuracy across all tasks, we observed that our approach exhibits significantly faster training speed compared to the baseline multi-tasking model (Fig. 5). Specifically, HCQNet trained for 100k iterations shows higher average accuracy than BaseNet-DPS trained for 300k iterations, showing more than three times faster training speed. This can be attributed to our query design, which explicitly leverages pre-computed high-level keypoints, thus enhancing training efficiency.

Qualitative comparison The effects of our proposed approach can be also found in qualitative results (Fig. 6). As shown in the second column of Fig. 6 (notice a horseman's foot), while task-specific method (Mask2Former) and naïve multi-tasking model (BaseNet-DPS) suffer from segmenting an occluded object, our model (HCQNet) can produce better segmentation result by jointly utilizing the information of other tasks encoded in our HCQ.

4.2 Comparison

To show our approach's effectiveness, we compare it with various task-specific models [2, 4, 5, 15, 22] and multi-task models [1, 8, 27] that partially address the combination of object detection, instance segmentation, and human pose estimation tasks. We use a residual network [7] (R-50 and R-152), a feature pyramid network [14] (fpn), and Swin-Transformer [16] (Swin-B) as backbone models, pretrained on the ImageNet-1K dataset [6]. Since no approach addresses all three tasks together, we divide the tasks into two separate groups based on evaluation protocols for fair comparison. Specifically, when evaluating tasks that include pose

Model	Backbone	Accuracy (mAP)		
		Det.	Pose.	Seg.
Mask R-CNN* [8]	R-50	52.0	✗	43.6
DETR* [2]	R-50	52.8	✗	✗
DAB-DETR* [15]	R-50	51.8	✗	✗
HCQNet	R-50	52.6	60.4	49.1
Mask2Former [6]	Swin-B	✗	✗	52.0
DAB-DETR [15]	Swin-B	56.4	✗	✗
Baseline-DPS	Swin-B	53.5	55.7	49.3
HCQNet	Swin-B	56.1	64.4	51.7

Model	Backbone	Accuracy (mAP)		
		Det.	Pose.	Seg.
Pose2Seg [27]	R-50-fpn	✗	59.9†	55.5
Pose2Seg(GT kpt)	R-50-fpn	✗	GT	58.2
PosePlusSeg [1]	R-152	✗	74.4	56.3
PETR [22]	R-50	✗	67.4	✗
HCQNet	Swin-B	68.9	65.2	65.1
HCQNet$_{ft}$	Swin-B	69.2	65.6	65.5

Table 2: Comparison with state-of-the-art task specific models on the COCO 2017 Person *minval* set **(left)** and the same set without small person instances **(right)**. The asterisk * denotes models trained for handling general classes, downloaded from the Detectron2 [24] and the authors' websites. † Pose2Seg uses a stand-alone model for pose estimation [20].

Model	Module	Cost (BFlops)	Prop. (%)
HCQNet	Backbone	363	69.96
	Pixel decoder	143	27.56
	Trans. decoder	12.60	2.43
	Class	0.0000512	0.00
	Mask	0.020634	0.00
	Box	0.128	0.02
	Pose	0.135	0.03
	Total	521	100

Model	Module	Cost (BFlops)	Prop.* (%)
Separate task-specific models	DAB-DETR [15]	361	69
	Mask2Former [6]	535	102
	PETR [22]	747	143
	Total	1643	315

*Proportion values are based on our HCQNet as the reference.

Table 3: Computational cost analysis. Backbone of all the models is Swin-B [16].

estimation, we exclude instances with a small size ($< 32 \times 32$ pixels) when computing mAP, as done in [1, 22, 27]. Note that once our HCQNet is jointly trained for all tasks, we use it for evaluation with both types of protocols.

In Table 2 (left), we compare our HCQNet with detection and segmentation models [2, 6, 8, 15]. For a fair comparison, we retrain state-of-the-art task specific models with the Swin-B backbone for a human class only; DAB-DETR [15] in object detection and Mask2Former [6] in instance segmentation. Table 2 shows that, even though our model runs three tasks simultaneously, it achieves comparable performance to task-specific state-of-the-art models.

In Table 2 (right), we compare models handling tasks including pose estimation [1, 22, 27]. In this experiment, we exclude small-sized human instances in the validation set as mentioned above. Our model shows a significantly higher segmentation accuracy than them (Table 2). Our pose estimation accuracy is lower than previous methods, partly because our models consider small objects in data augmentation, while pose estimation methods do not. Additionally, our model's simpler pose-specific head architecture and lack of specialization for human pose estimation contribute to its lower accuracy compared to state-of-the-art methods [1, 22]. We found that when we fine-tune our model on a training set focusing on larger instances, overall performance improves, indicating potential for further enhancement. We refer the reader to our supplementary material for results on the OCHuman dataset [27].

Computational cost analysis In earlier experiments, we show that our query design enables a unified network to perform multiple task harmoniously in terms of accuracy. Here, we evaluate the cost-effectiveness of our multi-task network. Computational costs are calculated for a 1024×1024 input image. Our network shares most computations in the backbone, pixel decoder (corresponding transformer encoder), and transformer decoder (Table 3). Task head overhead is minimal, demonstrating the scalability of our approach when increasing the number of target tasks. Notably, our model is more cost-efficient than executing task-specific state-of-the-art models separately.

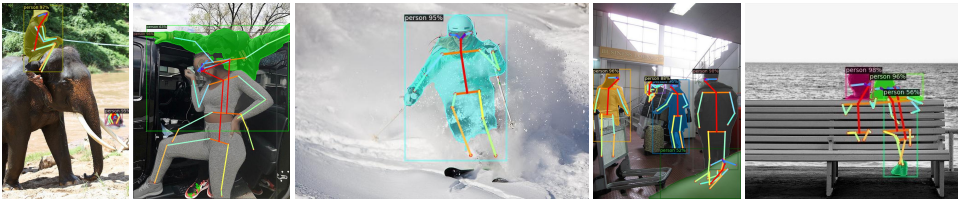

Figure 7: Failure cases due to the different natures of target tasks.

5 Discussion

Limitation & future work In this paper, we present a compact multi-task architecture in which multiple tasks fully utilize information within our unified representation, and their task-specific heads spend minimal overheads. To this end, our main focus lies in a unified query representation that can effectively encode multi-task information. Therefore, interactions between tasks through mechanisms such as loss balancing, inter-task-correlated loss, and other task-specific designs are less explored. Besides, our approach can be used in parallel with other advanced techniques such as query denoising [10] and top-down structures including prior information and hierarchical heuristics. While our straightforward design exhibits performance comparable to that of task-specific models, we believe that further exploration of these aspects could lead to improvements in our approach.

One specific direction involves considering the distinct natures of target tasks. Pose estimation, detection, and segmentation tasks can support each other in high-level perception, but differing task objectives can introduce ambiguity. For instance, pose estimation must handle occluded regions, unlike segmentation and detection, which deal with visible areas. As shown in Fig. 7, occluded pose joints can expand the bounding box, leading to a decline in object detection accuracy because the GT bbox is determined by enclosing the visible segments of the instance. Also, all pose joints are computed for any partial human instance (the rightmost column in Fig. 7). This may affect other tasks negatively although the corresponding bbox and mask of the instance do not always cover the pose. Considering the visibility of pose joints may alleviate this problem. Exploring potential mismatches among multiple tasks can offer valuable insights for improving the overall performance of multi-task learning.

In addition, due to the lack of labeled datasets for multiple tasks, including human pose estimation, we report performance on only two datasets in this paper. In theory, our approach would have a greater potential for improvement of multi-task learning as the number of target tasks increases. Constructing more datasets for the joint learning of current tasks and new human-related tasks such as person tracking/re-ID and action recognition would be an interesting future direction. Furthermore, while we focus on handling the human class only in this paper, our query design is not limited to a human class. Learnable keypoints are currently trained in a supervised manner. To deal with more general classes, we can consider manual labeling or unsupervised keypoint learning as future work.

Conclusion We introduce a novel human-centric representation for multi-task learning. In this approach, we develop a new query that carries the positional coordinates of keypoints, effectively capturing the structural information of human instances. To exploit the pre-computed high-level information within these queries, we employ learnable keypoints as conditional input for the task-specific heads and also combine them with deformable attention. Consequently, our proposed model demonstrates comparable performance to task-specific state-of-the-art models for various human recognition tasks, such as pose estimation, segmentation, and detection, while significantly reducing computational resource demands.

References

[1] Niaz Ahmad, Jawad Khan, Jeremy Yuhyun Kim, and Youngmoon Lee. Joint human pose estimation and instance segmentation with PosePlusSeg. In *Proc. AAAI*, 2022.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. ECCV*, 2020.

[3] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proc. CVPR*, 2018.

[4] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Proc. NeurIPS*, 2021.

[5] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proc. CVPR*, 2022.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proc. CVPR*, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Pro. ICCV*, 2017.

[9] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proc. CVPR*, 2019.

[10] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-DETR: Accelerate detr training by introducing query denoising. In *Proc. CVPR*, 2022.

[11] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask DINO: Towards a unified transformer-based framework for object detection and segmentation. In *Proc. CVPR*, 2023.

[12] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Proc. ICCV*, 2021.

[13] 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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proc. ECCV*, 2014.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017.

[15] 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. In *Proc. ICLR*, 2022.

[16] 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 *Proc. ICCV*, 2021.

[17] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Proc. NeurIPS*, 2020.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. ICLR*, 2019.

[19] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proc. ICCV*, 2021.

[20] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Proc. NeurIPS*, 2017.

[21] 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 *Proc. ECCV*, pages 269–286, 2018.

[22] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proc. CVPR*, 2022.

[23] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query design for transformer-based detector. In *Proc. AAAI*, 2022.

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

[25] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proc. CVPR*, 2018.

[26] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *Proc. ICLR*, 2023.

[27] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proc. CVPR*, 2019.

[28] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *Proc. NeurIPS*, 2021.

[29] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *Proc. ICLR*, 2021.