Appendix

A Experiments on Object Detection

A.1 Implementation Details

Datasets We conduct object detection experiments on MS COCO [12], which is a publicly available large-scale detection benchmark. MS COCO contains 118K images in the training set and 5K images in the validation set. All instances in these images belong to 80 different categories. The performance of a detector is evaluated with mean Average Precision (mAP) of all categories.

Hyperparameters During Training, Prompted Visual Indicator takes all 80 categories as inputs, and the retention policy retains $K' = 20$ categories. The indicator then generates $N = 100$ object queries for each of these remaining categories. As to the losses, we implement Focal Loss [11] for objectness branch, and a combination of ℓ_1 loss and the generalized IoU loss [19] for box regression in Sequence Predictor. The loss weights for them are 2, 5, 2 separately, which are similar to Deformable-DETR [30]. As to the auxiliary asymmetric loss [20] in Prompt Visual Indicator, the loss weight is set to be 0.25 for a comparable loss scale.

Training configuration Our training configuration directly follows Deformable-DETR [30]. Obj2Seq takes 16 images as a training batch. It is trained with an AdamW optimizer [15] for 50 epochs, with $\beta_1 = 0.9, \beta_2 = 0.999$ and weight decay 1×10^{-4}. The initial learning rate is 2×10^{-4}, and it decays by 0.1 after the 40th epoch. As to the input images, we apply scale augmentation and scale augmentation as in [2, 30]. We train Obj2Seq with 16 Nvidia V100 GPUs.

A.2 Further Improvements on Transformer Structure

Table 5: Experiments on iterative box refinement

Model	mAP	$AP50$	$AP75$	AP_S	AP_M	AP_L
Deformable DETR-R50 [30]	44.5	63.5	48.8	27.1	47.6	59.6
+ iterative box refinement [30]	46.3	65.0	50.1	28.3	49.2	61.5
Obj2Seq	45.7	64.8	49.5	28.0	48.8	60.2
+iterative box refinement	46.7	65.2	50.5	28.6	49.9	62.6

Currently, there emerge many variations of the visual transformer decoder with more efficient attention mechanisms or improved position embeddings. Though Obj2Seq mainly focuses on the unified framework that can process different tasks, it can also benefit from advanced structure modifications. Here we take a trail of iterative box refinement as in [30]. In Table 5, Obj2Seq with box refinement achieves 46.7%. It is +1.0% higher than the basic version, and also outperforms Deformable DETR with the same decoder structure by +0.4%. This indicates that iterative box refine is also available for Obj2Seq, and we are willing to test with more up-to-date improvements.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

B More comparison on Multi-Label Classification

As in Section 4.2, Obj2Seq trained for object detection can also be applied to predict the existence of different categories. Therefore, it has the ability to perform multi-label classification. Since object detection always has a larger input resolution (800×1333), the result is not that convincible. Here we evaluate with a smaller resolution, and provide a more detailed comparison with some other multi-label classification algorithms in Table 6. Obj2Seq achieves comparable performance with most methods. These results indicate that Obj2Seq can be extended to different usages.

Table 6: Experiments on COCO multi-label-classification. † indicates that this model is trained on ImageNet-21K, and ‡ on OpenImage.

Method	Backbone	Resolution	mAP_M
MCAR [7]	ResNet101	448×448	83.8
MCAR [7]	ResNet101	576×576	84.5
Query2Label [13]	ResNet101	448×448	84.9
ASL [13]	TResNet$_L$	448×448	86.6
Obj2Seq	ResNet50	480×640	87.0
Obj2Seq	ResNet50	800×1333	89.0
MlTr-XL† [5]		384×384	90.0
Query2Label† [13]	TResNet$_L$	640×640	90.3
ML-Decoder‡ [21]	TResNet$_L$	640×640	91.1

C Experiments on Human Pose Estimation

C.1 Implementation Details

Datasets We also conduct experiments on MS COCO [12], but with keypoint annotataions. We filter images in the training set, and retain the 4.8K images with at least 10 valid keypoint annotations as in [8]. As to the evaluation metric, we utilize Object Keypoint Similarity (OKS).

Model structure Our model consists of Image Feature Encoder, Object Transformer Decoder and General Sequence Predictor. Object Transformer Decoder consists of 6 decoder layers and 100 object queries. Sequence Predictor predicts 38 attributes for each object query. After predicting x, y, w, h, it continues to predict $\delta x, \delta y$ for all 17 keypoints. These attributes represent the offsets of these keypoints.

Losses and training configurations We implement binary cross entropy for the objectness branch, a combination of ℓ_1 loss and the generalized IoU loss [19] for box regression and a combination of ℓ_1 loss and oks loss [16] for keypoint offset prediction. All losses are normalized by the number of instances in ground truth, except that ℓ_1 for offsets is normalized by the number of valid keypoint annotations. They loss weights are 2, 5, 2, 40, 5 separately. The training scheduler is almost the same as in object detection. The only difference is that we construct a batch with 32 images, and the initial learning rate is set to 3×10^{-4}. When training with a 150-epoch schedule, we drop the learning rate by 0.1 after the 120th epoch instead. We train Obj2Seq for human pose estimation with 16 Nvidia V100 GPUs.

C.2 More comparisons

Here we provide a more complete table that lists most of the popular human pose estimation algorithms, including bottom-up methods, top-down methods with crop operation and top-down methods in an end-to-end way. Obj2Seq outperforms most of end-to-end top-down algorithms, and achieves a comparable performance with bottom-up ones. However, it still falls behind a lot when comparing with crop-based algorithms. This difference mainly attributes to the crop operation. This operation allows the model to extract related features from the exact position of each person. In order to achieve better results, some specific design for human pose estimation may benefit, which is beyond the topic of this paper.

Table 7: Human pose estimation results on MS COCO val. Results with ‡ indicates they utilize a pre-trained detector.

Method	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_M	AP_L	AR
Bottom-up methods								
OpenPose [1]	ResNet-Inception	-	61.8	84.9	67.5	57.1	68.2	-
AE-R50 [6]	ResNet 50	300	47.9	75.7	48.7	47.6	47.8	56.6
PersonLab [17]	ResNet 101	-	66.5	**86.2**	**71.9**	62.3	73.2	**70.7**
Pifpaf [9]	ResNet 101	75	66.7	-	-	**62.4**	72.9	-
HigherHRNet[4]	HigherHRNet	300	**66.9**	-	-	61.0	**75.7**	-
Top-down with crop operation								
CPN† [3]	ResNet-Inception	-	72.1	91.4	80.0	68.7	77.2	78.5
PRTR† [10]	HRNet-W32	-	72.1	90.4	79.6	68.1	79.0	79.4
SimpleBaseline† [26]	ResNet-152	-	73.7	91.9	81.1	70.3	80.0	79.0
HRNet† [23]	HRNet-W48	-	75.5	**92.5**	83.3	71.9	81.5	80.5
DARK† [27]	HRNet-W48	-	**76.2**	**92.5**	**83.6**	**72.5**	**82.4**	**81.1**
Top-Down with end-to-end frameworks								
Mask-RCNN [8]	ResNet50	-	63.1	**87.3**	68.7	57.8	71.4	-
CenterNet [29]	Hourglass-104	150	63.0	86.8	69.6	58.9	70.4	-
DirectPose [24]	ResNet50	100	63.1	85.6	68.8	57.7	71.3	-
PRTR‡ [10]	HRNet-W48	-	64.9	87.0	71.7	**60.2**	72.5	**74.1**
POET [22]	ResNet50	250	53.6	82.2	57.6	42.5	68.1	61.4
baseline	ResNet50	50	57.2	83.3	63.7	51.5	66.3	65.7
Obj2Seq	ResNet50	50	60.1	83.9	66.2	54.1	69.5	68.0
Obj2Seq	ResNet50	150	**65.0**	86.5	**71.8**	59.6	**74.0**	72.7

C.3 Effect of General Sequence Predictor on multi-task training

In order to demonstrate the performance of the sequence output format under multi-task training, we provide both metrics for human detection and pose estimation in this section. These experiments are conducted with three different multi-task prediction heads in Table 8.

Baseline utilizes 2 separate MLPs ($4d$ and $34d$) to predict detection and keypoint results from the processed object tokens. We follow previous DETRs to use 3-layer MLPs here.

2-Token combines the object queries with two additional task embeddings for detection and pose estimation, and obtains two task-specific tokens for each object. These queries are then fed into a transformer layer, with a self-attention between tokens of different tasks and a cross-attention layer inside. After that, MLPs are utilized to predict results for each task with corresponding task-specific tokens.

Table 8: Experiments on both human detection and pose estimation.

Method	Epochs	Human detection			Pose estimation		
		AP^{det}	AP^{det}_{50}	AP^{det}_{75}	AP^{kps}	AP^{kps}_{50}	AP^{kps}_{75}
Baseline	50	53.7	78.6	58.9	57.2	83.3	63.7
2-Token	50	53.9	80.2	58.4	58.3	83.7	64.9
Obj2Seq	50	**54.4**	**80.3**	**59.4**	**60.5**	**83.9**	**67.3**

We conduct experiments with the same 50-epoch training schedule. With the help of task embeddings, 2-Token head achieves higher performance than the simple MLP baseline. It makes use of self-attention among different tasks to enhance the performance. However, since Obj2Seq takes definite outputs from previous steps and utilizes them as inputs for subsequent steps, it is able to capture more explicit intra- and inter-task relations. Therefore, Obj2Seq achieves even better results. Moreover, this unified sequence format is consistent with text and audio tasks. It is more friendly to be extended for other multi-model applications.

D Code Details and Licenses

D.1 Details

Here we provide detailed algorithm for better understanding how Obj2Seq works. We mainly elaborate on two aspects. The first is to formulate the function of Prompted Visual Indicator in the pseudo code in Algorithm 1. The second is to demonstrate how the output logits z_t in Eq. (2) are transformed into final outputs exactly. We take object detection and human pose estimation as examples in Algorithm 2

Algorithm 1 Prompted Visual Indicator

Input: Image Feature $\boldsymbol{F}_I$, class prompts $\boldsymbol{C} = [\boldsymbol{c}^{(1)}...\boldsymbol{c}^{(K)}]$.

Output: Scores for class existence $s_c^{(k)}$, generated object queries $\hat{\boldsymbol{F}}_O$.

Params: Numer of prompt blocks $N_B = 2$, prompt blocks B_i, linear layers in classifier $Linear$, number of input classes K, number of retained classes K', number of object queries per class N, pos emb for object queries $\boldsymbol{P} = [p^{(1)}, ..., p^{(N)}]$.

Initialize class vectors with class prompts. $\boldsymbol{F}_C = \boldsymbol{C}$.
for $i = 1, ..., N_B$ **do**
    Extract class-related features from the image. $\boldsymbol{F}_C = B_i(\boldsymbol{F}_C, \boldsymbol{F}_I)$.
Calculate score for each class. $s_C^{(k)} = sigmoid(Linear(\boldsymbol{f}_c^{(k)}) \cdot \boldsymbol{c}^{(k)}/\sqrt{d}), (1 \le k \le K)$.
Retain K' classes according to the policy. $r_{k'}$ are indexes of retained classes, $k' = 1, ..., K'$.
Generate $K'N$ object queries. For each object query, $\hat{\boldsymbol{f}}_o^{(i)} = \boldsymbol{f}_c^{(r_{i//N})} + \boldsymbol{p}^{(i \mod N)}$.
return Scores for classes $s_C^{(k)}, k = 1, ..., K$. Initialized object queries $\hat{\boldsymbol{F}}_O$.

Algorithm 2 Non-parametric postprocess for object detection and pose estimation. Gray parts implemented for pose estimation only.

Input: Reference point (r_x, r_y), output logits $z_{1:T}$.
(T=4 for object detection; T=38 for pose estimation)

Output: The bounding box(x_c, y_c, w, h), keypoint offsets $(x_i, y_i), i = 1, ..., 17$.

Notations: $\mathcal{S}$ represents sigmoid function, $\mathcal{S}^{-1}$ represents inverse sigmoid function.

Calculate the bounding box.
$x_c = \mathcal{S}(\mathcal{S}^{-1}(r_x) + z_1)$.
$y_c = \mathcal{S}(\mathcal{S}^{-1}(r_y) + z_2)$.
$w = \mathcal{S}(z_3)$.
$h = \mathcal{S}(z_4)$.
Calculate keypoint coordinates.
for $i = 1, ..., 17$ **do**
    $x_i = x_c + w \times z_{2i+3}$.
    $y_i = y_c + h \times z_{2i+4}$.
return The bounding box (x_c, y_c, w, h), keypoint coordinates (x_i, y_i).

D.2 Licenses

The code for Obj2Seq is attached in the supplymentary meterial, and it will also be released to public later. Our code base is constructed mainly based on DETR [2], Deforamble DETR [30] and Anchor DETR [25]. For all code assets we refer to and their licenses, please see Table 9. In addition, we have also listed references at the beginning of each file in our code.

References

[1] Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7291–7299 (2017)

Table 9: Reference code assets and their licenses.

Code asset	License	Utility
DETR [2]	Apache 2.0	Foundation for our code
Deformable DETR [30]	Apache 2.0	Foundation for our code
Anchor DETR [25]	Apache 2.0	Foundation for our code
ASL [20]	MIT	Asymmetric loss for classification
Query2Label [13]	MIT	Metric for classification
Detic [28]	Apache 2.0	Generate CLIP-initialized vectors
CLIP [18]	MIT	Generate CLIP-initialized vectors
Mask R-CNN [8]	MIT	Dataset for keypoint annotations
Swin Transformer [14]	MIT	Config file

[2] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)

[3] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7103–7112 (2018)

[4] Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L.: Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5386–5395 (2020)

[5] Cheng, X., Lin, H., Wu, X., Yang, F., Shen, D., Wang, Z., Shi, N., Liu, H.: Mltr: Multi-label classification with transformer. arXiv preprint arXiv:2106.06195 (2021)

[6] Contributors, M.: Openmmlab pose estimation toolbox and benchmark. `https://github.com/open-mmlab/mmpose` (2020)

[7] Gao, B.B., Zhou, H.Y.: Learning to discover multi-class attentional regions for multi-label image recognition. IEEE Transactions on Image Processing **30**, 5920–5932 (2021)

[8] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)

[9] Kreiss, S., Bertoni, L., Alahi, A.: Pifpaf: Composite fields for human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11977–11986 (2019)

[10] Li, K., Wang, S., Zhang, X., Xu, Y., Xu, W., Tu, Z.: Pose recognition with cascade transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1944–1953 (2021)

[11] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

[12] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)

[13] Liu, S., Zhang, L., Yang, X., Su, H., Zhu, J.: Query2label: A simple transformer way to multi-label classification (2021)

[14] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)

[15] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[16] Maji, D., Nagori, S., Mathew, M., Poddar, D.: Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. arXiv preprint arXiv:2204.06806 (2022)

[17] Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: 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). pp. 269–286 (2018)

[18] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)

[19] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 658–666 (2019)

[20] Ridnik, T., Ben-Baruch, E., Zamir, N., Noy, A., Friedman, I., Protter, M., Zelnik-Manor, L.: Asymmetric loss for multi-label classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 82–91 (2021)

[21] Ridnik, T., Sharir, G., Ben-Cohen, A., Ben-Baruch, E., Noy, A.: Ml-decoder: Scalable and versatile classification head. arXiv preprint arXiv:2111.12933 (2021)

[22] Stoffl, L., Vidal, M., Mathis, A.: End-to-end trainable multi-instance pose estimation with transformers. arXiv preprint arXiv:2103.12115 (2021)

[23] Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5693–5703 (2019)

[24] Tian, Z., Chen, H., Shen, C.: Directpose: Direct end-to-end multi-person pose estimation. arXiv preprint arXiv:1911.07451 (2019)

[25] Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based detector. arXiv preprint arXiv:2109.07107 (2021)

[26] Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV). pp. 466–481 (2018)

[27] Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C.: Distribution-aware coordinate representation for human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7093–7102 (2020)

[28] Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: arXiv preprint arXiv:2201.02605 (2021)

[29] Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)

[30] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)