Panoptic SegFormer: Delving Deeper into Panoptic Segmentation with Transformers

Zhiqi Li[1], Wenhai Wang[2], Enze Xie[3], Zhiding Yu[4],
Anima Anandkumar[4,5], Jose M. Alvarez[4], Ping Luo[3], Tong Lu[1]

[1]Nanjing University [2]Shanghai AI Laboratory [3]The University of Hong Kong [4]NVIDIA [5]Caltech

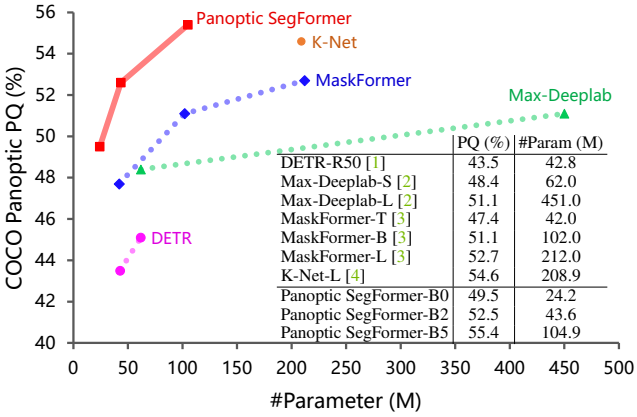
lzq@smail.nju.edu.cn wangwenhai@pjlab.org.cn xieenze@hku.hk zhidingy@nvidia.com
aanandkumar@nvidia.com josea@nvidia.com pluo@cs.hku.hk lutong@nju.edu.cn

Abstract

Panoptic segmentation involves a combination of joint semantic segmentation and instance segmentation, where image contents are divided into two types: things and stuff. We present Panoptic SegFormer, a general framework for panoptic segmentation with transformers. It contains three innovative components: an efficient deeply-supervised mask decoder, a query decoupling strategy, and an improved post-processing method. We also use Deformable DETR to efficiently process multi-scale features, which is a fast and efficient version of DETR. Specifically, we supervise the attention modules in the mask decoder in a layer-wise manner. This deep supervision strategy lets the attention modules quickly focus on meaningful semantic regions. It improves performance and reduces the number of required training epochs by half compared to Deformable DETR. Our query decoupling strategy decouples the responsibilities of the query set and avoids mutual interference between things and stuff. In addition, our post-processing strategy improves performance without additional costs by jointly considering classification and segmentation qualities to resolve conflicting mask overlaps. Our approach increases the accuracy 6.2% PQ over the baseline DETR model. Panoptic SegFormer achieves state-of-the-art results on COCO test-dev with 56.2% PQ. It also shows stronger zero-shot robustness over existing methods.

1. Introduction

Semantic segmentation and instance segmentation are two important and related vision tasks. Their underlying connections recently motivated panoptic segmentation as a unification of both the tasks [6]. In panoptic segmentation, image contents are divided into two types: things and stuff. Things refer to countable instances (*e.g.*, person, car) and each instance has a unique id to distinguish it from the other instances. Stuff refers to the amorphous and uncountable regions (*e.g.*, sky, grassland) and has no instance id [6].



	PQ (%)	#Param (M)
DETR-R50 [1]	43.5	42.8
Max-Deeplab-S [2]	48.4	62.0
Max-Deeplab-L [2]	51.1	451.0
MaskFormer-T [3]	47.4	42.0
MaskFormer-B [3]	51.1	102.0
MaskFormer-L [3]	52.7	212.0
K-Net-L [4]	54.6	208.9
Panoptic SegFormer-B0	49.5	24.2
Panoptic SegFormer-B2	52.5	43.6
Panoptic SegFormer-B5	55.4	104.9

Figure 1. Comparison to the prior arts in panoptic segmentation methods on the COCO val2017 split. Panoptic SegFormer models outperform the other counterparts among different models. Panoptic SegFormer (PVTv2-B5 [5]) achieves 55.4% PQ, surpassing previous methods with significantly fewer parameters.

Recent works [1–3] attempt to employ transformers to handle both things and stuff through a query set. For example, DETR [1] simplifies the workflow of panoptic segmentation by adding a panoptic head on top of an end-to-end object detector. Unlike previous methods [6, 7], DETR does not require additional handcrafted pipelines [8, 9]. While being simple, DETR also causes some issues: (1) It requires a lengthy training process to converge; (2) Because the computational complexity of self-attention is squared with the length of the input sequence, the feature resolution of DETR is limited. So that it uses an FPN-style [1, 10] panoptic head to generate masks, which always suffer low-fidelity boundaries; (3) It handles things and stuff equally, yet representing them with bounding boxes, which may be suboptimal for stuff [2, 3]. Although DETR achieves excellent performance on the object detection task, its superiority on panoptic segmentation has not been well demonstrated. In order

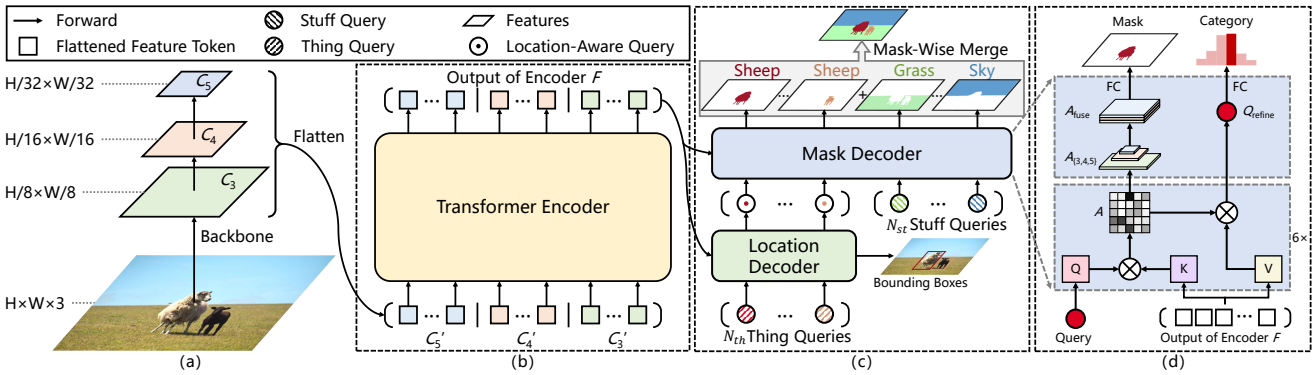


Figure 2. **Overview of Panoptic SegFormer.** Panoptic SegFormer is composed of backbone, encoder, and decoder. The backbone and the encoder output and refine multi-scale features. Inputs of the location decoder are N_{th} thing queries and the multi-scale features. We feed N_{th} thing queries from the location decoder and N_{st} stuff queries to the mask decoder. The location decoder aims to learn reference points of queries, and the mask decoder predicts the final category and mask. Details of the decoder will be introduced below. We use a mask-wise merging method instead of the commonly used pixel-wise argmax method to perform inference.

to overcome the defects of DETR on panoptic segmentation, we propose a series of novel and effective strategies that improve the performance of transformer-based panoptic segmentation models by a large margin.

Our approach. In this work, we propose Panoptic SegFormer, a concise and effective framework for panoptic segmentation with transformers. Our framework design is motivated by the following observations: 1) Deep supervision matters in learning high-qualities discriminative attention representations in the mask decoder. 2) Treating things and stuff with the same recipe [1] is suboptimal due to the different properties between things and stuff [6]. 3) Commonly used post-processing such as pixel-wise argmax [1–3] tends to generate false-positive results due to extreme anomalies. We overcome these challenges in Panoptic SegFormer framework as follows:

- We propose a mask decoder that utilizes multi-scale attention maps to generate high-fidelity masks. The mask decoder is deeply-supervised, promoting discriminative attention representations in the intermediate layers with better mask qualities and faster convergence.
- We propose a query decoupling strategy that decomposes the query set into a thing query set to match things via bipartite matching and another stuff query set to process stuff with class-fixed assign. This strategy avoids mutual interference between things and stuff within each query and significantly improves the qualities of stuff segmentation. Kindly refer to Sec. 3.3.1 and Fig. 3 for more details.
- We propose an improved post-processing method to generate results in panoptic format. Besides being more efficient than the widely used pixel-wise argmax method, our method contains a mask-wise merging strategy that considers both classification probability and predicted mask qualities. Our post-processing method alone renders a 1.3% PQ improvement to DETR [1].

We conduct extensive experiments on COCO [11] dataset. As shown in Fig. 1, Panoptic SegFormer significantly surpasses priors arts such as MaskFormer [3] and K-Net [4] with much fewer parameters. With deformable attention [12] and our deeply-supervised mask decoder, our method requires much fewer training epochs than previous transformer-based methods (24 *vs.* 300+) [1, 3]. In addition, our approach also achieves competitive performance with current methods [13, 14] on the instance segmentation task.

2. Related Work

Panoptic Segmentation. Panoptic segmentation becomes a popular task for holistic scene understanding [6, 15–17]. The panoptic segmentation literature mainly treats this problem as a joint task of instance segmentation and semantic segmentation where things and stuff are handled separately [18, 19]. Kirillov *et al.* [6] proposed the concept of and benchmark of panoptic segmentation together with a baseline that directly combines the outputs of individual instance segmentation and semantic segmentation models. Since then, models such as Panoptic FPN [7], UPSNet [9] and AUNet [20] have improved the accuracy and reduced the computational overhead by combining instance segmentation and semantic segmentation into a single model. However, these methods approximate the target task by solving the surrogate sub-tasks, therefore introducing undesired model complexities and suboptimal performance.

Recently, efforts have been made to unify the framework of panoptic segmentation. Li *et al.* [21] proposed Panoptic FCN where the panoptic segmentation pipeline is simplified with a "top-down meets bottom-up" two-branch design similar to CondInst [22]. In their work, things and stuff are jointly modeled by an object/region-level kernel branch and an image-level feature branch. Several recent works represent things and stuff as queries and perform end-

to-end panoptic segmentation via transformers. DETR [1] predicts the bounding boxes of things and stuff and combines the attention maps of the transformer decoder and the feature maps of ResNet [23] to perform panoptic segmentation. Max-Deeplab [2] directly predicts object categories and masks through a dual-path transformer regardless of the category being things or stuff. On top of DETR, MaskFomer [3] used an additional pixel decoder to refine high spatial resolution features and generated the masks by multiplying queries and features from the pixel decoder. Due to the computational complexity of self attention [24], both DETR and MaskFormer use feature maps with limited spatial resolutions for panoptic segmentation, which hurts the performance and requires combining additional high-resolution feature maps in final mask prediction. Unlike the methods mentioned above, our query decoupling strategy deals with things and stuff with separate query sets. Although thing and stuff queries are designed for different targets, they are processed by the mask decoder with the same workflow. Prediction results of these queries are in the same format so that we can process them in an equal manner during the post-processing procedure. One concurrent work [4] employs a similar line of thinking to use dynamic kernels to perform instance and semantic segmentation, and it aims to utilize unified kernels to handle various segmentation tasks. In contrast to it, we aim to delve deeper into the transformer-based panoptic segmentation. Due to the different nature of various tasks, whether a unified pipeline is suitable for these tasks is still an open problem. In this work, we utilize an additional location decoder to assist things to learn location clues and get better results.

End-to-end Object Detection. The recent popular end-to-end object detection frameworks have inspired many other related works [13, 25]. DETR [1] is arguably the most representative end-to-end object detector among these methods. DETR models the object detection task as a dictionary lookup problem with learnable queries and employs an encoder-decoder transformer to predict bounding boxes without extra post-processing. DETR greatly simplifies the conventional detection framework and removes many hand-crafted components such as Non-Maximum Suppression (NMS) [26, 27] and anchors [27]. Zhu *et al*. [12] proposed Deformable DETR, which further reduces the memory and computational cost through deformable attention layers. In this work, we adopt deformable attention [12] for the improved efficiency and convergence over DETR [1].

3. Methods

3.1. Overall Architecture

As illustrated in Fig. 2, Panoptic SegFormer consists of three key modules: transformer encoder, location decoder, and mask decoder, where (1) the transformer encoder is applied to refine the multi-scale feature maps given by the backbone, (2) the location decoder is designed to capturing location clues of things, and (3) the mask decoder is for final classification and segmentation.

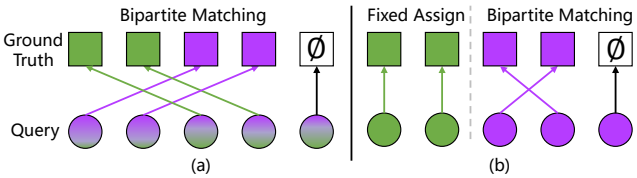


Figure 3. (a) Methods [1–3] adopt one query set to match things (purple squares) and stuff (green squares) jointly. (b) We use one thing query set (purple circles) to target things through bipartite matching and one stuff query set (green circles) to predict stuff by a class-fixed assign strategy. ∅ is assigned to not-matched queries.

Our architecture feeds an input image $X \in \mathbb{R}^{H\times W\times 3}$ to the backbone network, and obtains the feature maps C_3, C_4, and C_5 from the last three stages, of which the resolutions are $1/8$, $1/16$ and $1/32$ compared to the input image, respectively. We project the three feature maps to the ones with 256 channels by a fully-connected (FC) layer, and flatten them into feature tokens C_3', C_4', and C_5'. Here, we define L_i as $\frac{H}{2^{i+2}} \times \frac{W}{2^{i+2}}$, and the shapes of C_3', C_4', and C_5' are $L_1 \times 256$, $L_2 \times 256$, and $L_3 \times 256$, respectively. Next, using the concatenated feature tokens as input, the transformer encoder outputs the refined features of size $(L_1{+}L_2{+}L_3){\times}256$. After that, we use N_{th} and N_{st} randomly initialized things and stuff queries to describe things and stuff separately. Location decoder refines N_{th} thing queries by detecting the bounding boxes of things to capture location information. The mask decoder then takes both things and stuff queries as input and predicts mask and category at each layer.

During inference, we adopt a mask-wise merging strategy to convert the predicted masks from final mask decoder layer into the panoptic segmentation results, which will be introduced in detail in Sec. 3.5.

3.2. Transformer Encoder

High-resolution and the multi-scale features maps are important for the segmentation tasks [7, 21, 28]. Since the high computational cost of self-attention layer, previous transformer-based methods [1, 3] can only process low-resolution feature maps (*e.g*., ResNet C_5) in their encoders, which limits the segmentation performance. Different from these methods, we employ the deformable attention [12] to implement our transformer encoder. Due to the low computational complexity of the deformable attention, our encoder can refine and involve positional encoding [24] to high-resolution and multi-scale feature maps F.

3.3. Decoder

In this section, we introduce our query decoupling strategy firstly, and then we will explain the details of our location decoder and mask decoder.

3.3.1 Query Decoupling Strategy

We argue that using one query set to handle both things and stuff equally is suboptimal. Since there many different properties between them, things and stuff is likely to interfere with each other and hurt the model performance, especially for PQ^{st}. To prevent things and stuff from interfering with each other, we apply a query decoupling strategy in Panoptic SegFormer, as shown in Fig. 3. Specifically, N_{th} thing queries are used to predict things results, and N_{st} stuff queries target stuff only. Using this form, things and stuff queries can share the same pipeline since they are in the same format. We can also customize private workflow for things or stuff according to the characteristics of different tasks. In this work, we use an additional location decoder to detect individual instances with thing queries, and this will assist in distinguishing between different instances [6]. Mask decoder accepts both thing queries and stuff queries and generates the final masks and categories. Note that, for thing queries, ground truths are assigned by bipartite matching strategy. For stuff, We use a class-fixed assign strategy, and each stuff query corresponds to one stuff category. Thing and stuff queries will output results in the same format, and we handle these results with a uniform post-processing method.

3.3.2 Location Decoder

Location information plays an important role in distinguishing things with different instance ids in the panoptic segmentation task [22, 28, 29]. Inspired by this, we employ a location decoder to introduce the location information of things into the learnable queries. Specifically, given N_{th} randomly initialized thing queries and the refined feature tokens generated by transformer encoder, the decoder will output N_{th} location-aware queries.

In the training phase, we apply an auxiliary MLP head on top of location-aware queries to predict the bounding boxes and categories of the target object, We supervise the prediction results with a detection loss $\mathcal{L}_{\text{det}}$. The MLP head is an auxiliary branch, which can be discarded during the inference phase. The location decoder follows Deformable DETR [12]. Notably, the location decoder can learn location information by predicting the mass centers of masks instead of bounding boxes. This box-free model can still achieve comparable results to our box-based model.

3.3.3 Mask Decoder

As shown in Fig. 2 (d), the mask decoder is proposed to predict the categories and masks according to the given queries. The queries Q of the mask decoder are the location-aware thing queries from the location decoder or the class-fixed stuff queries. The keys K and values V of the mask decoder are projected from the refined feature tokens F from the transformer encoder. We first pass thing queries through the mask decoder, and then fetch the attention map $A \in \mathbb{R}^{N\times h\times(L_1+L_2+L_3)}$ and the refined query $Q_{\text{refine}} \in \mathbb{R}^{N\times 256}$ from each decoder layer, where $N = N_{\text{th}}+N_{\text{st}}$ is the whole query number, h is the number of attention heads, and $L_1+L_2+L_3$ is the length of feature tokens F.

Similar to methods [1, 2], we directly perform classification through a FC layer on top of the refined query Q_{refine} from each decoder layer. Each thing query needs to predict probabilities over all thing categories. Stuff query only predicts the probability of its corresponding stuff category.

At the same time, to predict the masks, we first split and reshape the attention maps A into attention maps A_3, A_4, and A_5, which have the same spatial resolution as C_3, C_4, and C_5. This process can be formulated as:

$$(A_3, A_4, A_5) = \text{Split}(A), \quad A_i \in \mathbb{R}^{\frac{H}{2^{i+2}}\times\frac{W}{2^{i+2}}\times h}, \tag{1}$$

where $\text{Split}(\cdot)$ denotes the split and reshaping operation. After that, as illustrated in Eq. (2), we upsample these attention maps to the resolution of $H/8\times W/8$ and concatenate them along the channel dimension,

$$A_{\text{fused}} = \text{Concat}(A_1, \text{Up}_{\times 2}(A_2), \text{Up}_{\times 4}(A_3)), \tag{2}$$

where $\text{Up}_{\times 2}(\cdot)$ and $\text{Up}_{\times 4}(\cdot)$ mean the 2 times and 4 times bilinear interpolation operations, respectively. $\text{Concat}(\cdot)$ is the concatenation operation. Finally, based on the fused attention maps A_{fused}, we predict the binary mask through a 1×1 convolution.

Previous literature [12] argues that the reason for slow convergence of DETR is that attention modules equally pay attention to all the pixels in the feature maps, and learning to focus on sparse meaningful locations requires plenty of effort. We use two key designs to solve this problem in our mask decoder: (1) Using an ultra-light FC head to generate masks from the attention maps, ensuring attention modules can be guided by ground truth mask to learn where to focus on. This FC head only contains 200 parameters, which ensures the semantic information of attention maps is highly related to the mask. Intuitively, the ground truth mask is exactly the meaningful region on which we expect the attention module to focus. (2) We employ deep supervision in the mask decoder. Attention maps of each layer will be supervised by the mask, the attention module can capture meaningful information in the earlier stage. This can highly accelerate the learning process of attention modules.

3.4. Loss Function

During training, our overall loss function of Panoptic SegFormer can be written as:

$$\mathcal{L} = \lambda_{\text{things}}\mathcal{L}_{\text{things}} + \lambda_{\text{stuff}}\mathcal{L}_{\text{stuff}}, \tag{3}$$

where L_{things} and L_{stuff} are loss for things and stuff, separately. λ_{things} and λ_{stuff} are hyperparameters.

Things Loss. Following common practices [1, 30], we search the best bipartite matching between the prediction

set and the ground truth set. Specifically, we utilize Hungarian algorithm [31] to search for the permutation with the minimum matching cost, which is the sum of the classification loss $\mathcal{L}_{\text{cls}}$, detection loss $\mathcal{L}_{\text{det}}$ and the segmentation loss $\mathcal{L}_{\text{seg}}$. The overall loss function for the thing categories is accordingly defined as follows:

$$\mathcal{L}_{\text{things}} = \lambda_{\text{det}}\mathcal{L}_{\text{det}} + \sum_{i}^{D_m}(\lambda_{\text{cls}}\mathcal{L}^i_{\text{cls}} + \lambda_{\text{seg}}\mathcal{L}^i_{\text{seg}}), \quad (4)$$

where λ_{cls}, λ_{seg}, and λ_{loc} are the weights to balance three losses. D_m is the number of layers in the mask decoder. $\mathcal{L}^i_{\text{cls}}$ is the classification loss that is implemented by Focal loss [27], and $\mathcal{L}^i_{\text{seg}}$ is the segmentation loss implemented by Dice loss [32]. $\mathcal{L}_{\text{det}}$ is the loss of Deformable DETR that used to perform detection.

Stuff Loss. We use a fixed matching strategy for stuff. Thus there is a one-to-one mapping between stuff queries and stuff categories. The loss for the stuff categories is similarly defined as:

$$\mathcal{L}_{\text{stuff}} = \sum_{i}^{D_m}(\lambda_{\text{cls}}\mathcal{L}^i_{\text{cls}} + \lambda_{\text{seg}}\mathcal{L}^i_{\text{seg}}), \quad (5)$$

where $\mathcal{L}^i_{\text{cls}}$ and $\mathcal{L}^i_{\text{seg}}$ are the same as those in Eq. (4).

3.5. Mask-Wise Merging Inference

Panoptic Segmentation requires each pixel to be assigned a category label (or void) and instance id (ignored for stuff) [6]. One challenge of panoptic segmentation is that it requires generating non-overlap results. Recent methods [1–3] directly use pixel-wise argmax to determine the attribution of each pixel, and this can solve the overlap problem naturally. Although pixel-wise argmax strategy is simple and effective, we observe that it consistently produces false-positive results due to the abnormal pixel values.

Unlike pixel-wise argmax resolves conflicts on each pixel, we propose the mask-wise merging strategy by resolving the conflicts among predicted masks. Specifically, we use the confidence scores of masks to determine the attribution of the overlap region. Inspired by previous NMS methods [28], the confidence scores take into both classification probability and predicted mask qualities. The confidence score of the i-th result can be formulated as:

$$s_i = p_i^{\alpha} \times \text{average}(\mathbb{1}_{\{m_i[h,w]>0.5\}} m_i[h,w])^{\beta}, \quad (6)$$

where p_i is the most likely class probability of i-th result. $m_i[h,w]$ is the mask logit at pixel $[h,w]$, α, β are used to balance the weight of classification probability and segmentation qualities.

As illustrated in Algorithm 1, mask-wise merging strategy takes c, s, and m as input, denoting the predicted categories, confidence scores, and segmentation masks, respectively. It outputs a semantic mask `SemMsk` and an instance id mask `IdMsk`, to assign a category label and an instance id to each pixel. Specifically, `SemMsk` and `IdMsk` are first initialized by zeros. Then, we sort prediction results in descending order of confidence score and fill the sorted predicted masks into `SemMsk` and `IdMsk` in order. Then we discard the results with confidence scores below t_{cls} and remove the overlaps with lower confidence scores. Only remained non-overlap part with a sufficient fraction t_{keep} to origin mask will be kept. Finally, the category label and unique id of each mask are added to generate non-overlap panoptic format results.

Algorithm 1: Mask-Wise Merging

```
def MaskWiseMergeing(c,s,m):
    # category c ∈ R^N
    # confidence score s ∈ R^N
    # mask m ∈ R^{N×H×W}
    SemMsk = np.zeros(H,W)
    IdMsk = np.zeros(H,W)
    order = np.argsort(-s)
    id = 0
    for i in order:
        m_i = m[i]>0.5 & (SemMsk==0)
        if s[i]< t_cnf or m_i/m[i]>0.5 < t_keep:
            continue
        SemMsk[m_i] = c[i]
        IdMsk[m_i] = id
        id += 1
return SemMsk,IdMsk
```

4. Experiments

We evaluate Panoptic SegFormer on COCO [11] and ADE20K dataset [33], comparing it with several state-of-the-art methods. We provide the main results of panoptic segmentation and instance segmentation. We also conduct detailed ablation studies to verify the effects of each module. Please refer to Appendix for implementation details.

4.1. Dataset

We perform experiments on COCO 2017 datasets [11] without external data. The COCO dataset contains 118K training images and 5k validation images, and it contains 80 things and 53 stuff. We further demonstrate the generality of our model on the ADE20K dataset [33], which contains 100 things and 50 stuff.

4.2. Main Results

Panoptic segmentation. We conduct experiments on COCO `val` set and `test-dev` set. In Tab. 1 and Tab. 2, we report our main results, comparing with other state-of-the-art methods. Panoptic SegFormer attains 49.6% PQ on COCO `val` with ResNet-50 as the backbone and single-scale input, and it surpasses previous methods K-Net [4] and DETR [1] over 2.5% PQ and 6.2% PQ, respectively. Except for the remarkable performance, the training of

Method	Backbone	Epochs	PQ	PQ^{th}	PQ^{st}	#P	#F
Panoptic FPN [7]	R50	36	41.5	48.5	31.1	-	-
SOLOv2 [28]	R50	36	42.1	49.6	30.7	-	-
DETR [1]	R50	325	43.4	48.2	36.3	42.9	248
Panoptic FCN [21]	R50	36	43.6	49.3	35.0	37.0	244
K-Net [4]	R50	36	47.1	51.7	40.3	-	-
MaskFormer [3]	R50	300	46.5	51.0	39.8	45.0	181
Panoptic SegFormer	R50	**12**	48.0	52.3	41.5	51.0	214
Panoptic SegFormer	R50	24	**49.6**	**54.4**	**42.4**	51.0	214
DETR [1]	R101	325	45.1	50.5	37.0	61.8	306
Max-Deeplab-S [2]	Max-S [2]	54	48.4	53.0	41.5	61.9	162
MaskFormer [3]	R101	300	47.6	52.5	40.3	64.0	248
Panoptic SegFormer	R101	24	**50.6**	**55.5**	**43.2**	69.9	286
Max-Deeplab-L [2]	Max-L [2]	54	51.1	57.0	42.2	451.0	1846
Panoptic FCN [36]	Swin-L†	36	51.8	58.6	41.6	-	-
MaskFormer [3]	Swin-L†	300	52.7	58.5	44.0	212.0	792
K-Net [4]	Swin-L†	36	54.6	60.2	46.0	208.9	-
Panoptic SegFormer	Swin-L†	24	**55.8**	**61.7**	**46.9**	221.4	816
Panoptic SegFormer	PVTv2-B5†	24	55.4	61.2	46.6	104.9	349

Table 1. Experiments on COCO val set. #P and #F indicate number of parameters (M) and number of FLOPs (G). Panoptic SegFormer (R50) achieves 49.6% PQ on COCO `val`, surpassing previous methods such as DETR [1] and MaskFormer [3] over 6.2% PQ and 3.1% PQ respectively. † notes that backbones are pre-trained on ImageNet-22K.

Method	Backbone	PQ	PQ^{th}	PQ^{st}	SQ	RQ
Max-Deeplab-L [2]	Max-L [2]	51.3	57.2	42.4	82.5	61.3
Innovation [35]	ensemble	53.5	61.8	41.1	**83.4**	63.3
MaskFormer [3]	Swin-L†	53.3	59.1	44.5	82.0	64.1
K-Net [4]	Swin-L†	55.2	61.2	46.2	82.4	66.1
Panoptic SegFormer	R50	50.2	55.3	42.4	81.9	60.4
Panoptic SegFormer	R101	50.9	56.2	43.0	82.0	61.2
Panoptic SegFormer	Swin-L†	**56.2**	**62.3**	**47.0**	82.8	**67.1**
Panoptic SegFormer	PVTv2-B5†	55.8	61.9	46.5	83.0	66.5

Table 2. Experiments on COCO test-dev set. † notes that backbones are pre-trained on ImageNet-22K.

Method	Backbone	PQ	PQ^{th}	PQ^{st}	SQ	RQ
BGRNet [37]	R50	31.8	-	-	-	-
Auto-Panoptic [38]	ShuffleNetV2 [39]	32.4	-	-	-	-
MaskFormer [3]	R50	34.7	32.2	**39.7**	76.7	42.8
MaskFormer [3]	R101	35.7	34.5	38.0	77.4	43.8
Panoptic SegFormer	R50	**36.4**	**35.3**	38.6	**78.0**	**44.9**

Table 3. Panoptic segmentation results on ADE20K val set.

Panoptic SegFormer is efficient. Under 1× training strategy (12 epochs), Panoptic SegFormer (R50) achieves 48.0% PQ that outperforms MaskFormer [3] that training 300 epochs by 1.5% PQ. Enhanced by vision transformer backbone Swin-L [34], Panoptic SegFormer attains a new record of 56.2% PQ on COCO `test-dev` without bells and whistles, surpassing MaskFormer [3] over 2.9% PQ. Our method even surpasses the previous competition-level method Innovation [35] over 2.7 % PQ. We also obtain comparable performance by employing PVTv2-B5 [5], while the model parameters and FLOPs are reduced significantly compared to Swin-L. Panoptic SegFormer also outperforms MaskFormer by 1.7% PQ on ADE20K dataset [33], see Tab. 3.

Instance segmentation. Panoptic SegFormer can be converted to an instance segmentation model by just discarding stuff queries. In Tab. 4, we report our instance segmentation results on COCO `test-dev` set. We achieve results comparable to the current state-of-the-art methods such as QueryInst [13] and HTC [14], and 1.8 AP higher than K-Net [4]. Using random crops during training boosts the AP by 1.3 percentage points.

Method	Backbone	AP^{seg}	AP^{seg}_S	AP^{seg}_M	AP^{seg}_L
Mask R-CNN [40]	R50	37.5	21.1	39.6	48.3
SOLOv2 [28]	R50	38.8	16.5	41.7	56.2
K-Net [4]	R50	38.6	19.1	42.0	**57.7**
SOLQ [25]	R50	39.7	21.5	42.5	53.1
HTC [14]	R50	39.7	22.6	42.2	50.6
QueryInst [13]	R50	40.6	**23.4**	42.5	52.8
Ours (w/o crop)	R50	40.4	21.1	43.8	54.7
Ours (w/ crop)	R50	**41.7**	21.9	**45.3**	56.3

Table 4. Instance segmentation on COCO `test-dev` set.

	Epochs	PQ	#Params	FLOPs	FPS
baseline (DETR [1])	325	43.4	42.9M	247.5G	4.9
+ mask-wise merging	325	44.7	42.9M	247.5G	6.1
++ ms deformable attention [12]	50	47.3	44.9M	618.7G	2.7
+++ mask decoder	24	48.5	51.0M	214.8G	7.8
++++ query decoupling	24	49.6	51.0M	214.2G	7.8

Table 5. We increase the panoptic segmentation performance of DETR [1] (R50 [23]) from 43.4% PQ to 49.6% PQ with fewer training epochs, less computation cost, and faster inference speed.

4.3. Ablation Studies

First, we show the effect of each module in Tab. 5. Compared to baseline DETR, our model achieves better performance, faster inference speed and significantly reduces the training epochs. We use Panoptic SegFormer (R50) to perform ablation experiments by default.

Effect of Location Decoder. Location decoder assists queries to capture the location information of things. Tab. 6 shows the results with varying the number of layers in the location decoder. With fewer location decoder layers, our model performs worse on things, and it demonstrates that learning location clues through the location decoder is beneficial to the model to handle things better. * notes we predict mass centers rather than bounding boxes in our location decoder, and this box-free model achieves comparable results (49.2% PQ *vs*. 49.6% PQ).

#Layer	PQ	PQ^{th}	PQ^{st}
0	47.0	50.0	42.5
1	47.7	51.1	42.5
2	48.1	51.8	42.5
6* (box-free)	49.2	53.5	42.6
6	49.6	54.4	42.4

Table 6. Ablate location decoder.

Mask-wise Merging. As shows in Tab. 7, we compare our mask-wise merging strategy against pixel-wise argmax strategy on various models. We use both Mask PQ and Boundary PQ [41] to make our conclusions more credible. Models with mask-wise merging strategy always performs better. DETR with mask-wise merging outperforms origin DETR by 1.3% PQ [1]. In addition, our mask-wise merging is 20% less time-consuming than DETR's pixel-wise argmax since DETR uses more tricks in its code, such as merging stuff with the same category and iteratively remov-

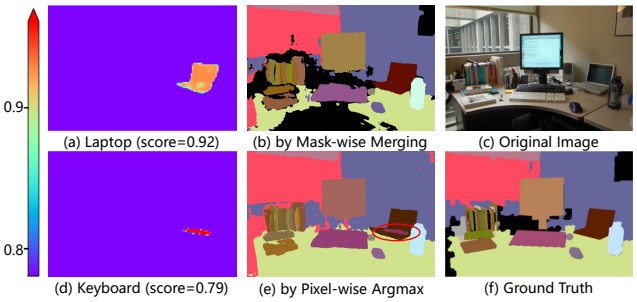


Figure 4. While using pixel-wise argmax, the keyboard is covered on the laptop (noted by the red circle in (e). However, the laptop has a higher classification probability than the keyboard. The pixel-wise argmax strategy fails to use this important clue. Masks logits were generated through DETR-R50 [1].

Method	Mask PQ			Boundary PQ [41]		
	PQ	SQ	RQ	PQ	SQ	RQ
DETR (p)	43.4	79.3	53.8	32.8	71.0	45.2
DETR (m)	44.7	80.2	54.7	33.7	71.1	46.5
D-DETR-MS (p)	46.3	80.0	56.5	37.1	72.1	50.2
D-DETR-MS (m)	47.3	81.1	56.8	38.0	72.3	51.0
MaskFormer (p)	45.6	80.2	55.8	-	-	-
MaskFormer (p*)	46.5	80.4	56.8	36.8	72.5	49.8
MaskFormer (m)	46.8	80.4	57.2	37.6	72.6	51.1
Panoptic SegFormer (p)	48.4	80.7	58.9	39.3	73.0	52.9
Panoptic SegFormer (m)	49.6	81.6	59.9	40.4	73.4	54.2

Table 7. Effect of mask-wise merging strategy. The table shows the results of models with different post-processing methods, and the backbone is ResNet-50. "(p)" refers to using pixel-wise argmax as the post-processing method. "(p*)" considers both class probability and mask prediction probability in its pixel-wise argmax strategy [3]. Models with "(m)" that employ mask-wise merging always perform better in both Mask PQ and Boundary PQ [41] than pixel-wise argmax method.

Method	PQ	PQ^{th}	PQ^{st}	AP^{box}	AP^{seg}
DETR [1]	43.4	48.2	36.3	38.8	31.1
D-DETR-MS [12]	47.3	52.6	39.0	45.3	37.6
Panoptic FCN [21]	43.6	49.3	35.0	36.6	34.5
Ours (Joint Matching)	48.5	**54.5**	39.5	44.1	37.7
Ours (Query Decoupling)	**49.6**	54.4	**42.4**	**45.6**	**39.5**

Table 8. Effect of query decoupling strategy. PQ and AP scores of various panoptic segmentation models on COCO `val2017`.

ing masks with small areas. Fig. 4 shows one typical fail case of using pixel-wise argmax.

Mask Decoder. Our proposed mask decoder converges faster since the ground truth masks guide the attention module to focus on meaningful regions. Fig. 5 shows the convergence curves of several models. We only supervise the last layer of the mask decoder while not employing deep supervision. We can observe that our method achieves 49.6% PQ with training for 24 epochs, and longer training has little effect. However, D-DETR-MS needs at least 50 epochs to achieve better performance. Deep supervision is vital for our mask decoder to perform better and converge faster. Fig. 6 shows the attention maps of different layers in the

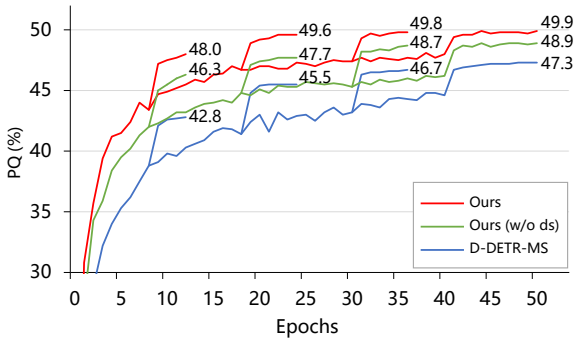


Figure 5. Convergence curves of Panoptic SegFormer and D-DETR-MS. We train models with different training schedules. "w/o ds" refers that we do not employ deep supervision in the mask decoder. The learning rate is reduced where the curves leap.

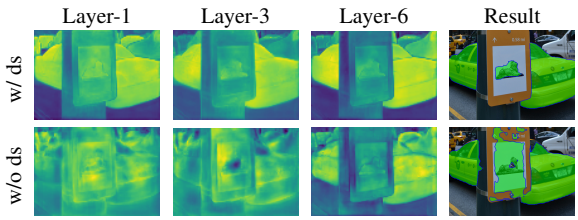


Figure 6. Attention maps of different layers in the mask decoder. "ds" refers to deep supervision.

mask decoder, and the attention module focuses on the target car in the previous layer when using deep supervision. The attention maps are very similar to the final predicted masks, since masks are generated by attention maps with a lightweight FC head.

Since our mask decoder can generate masks from each layer, we evaluate the performance of each layer in the mask decoder, see Tab. 10. During inference, using the first two layers of mask decoder will be on par with the whole mask decoder. It also inferences faster because the computational cost decreases. PQ^{th} is hardly affected by the number of layers, PQ^{st} performs a little poorly in the first layer. The reason is that the location decoder has made additional refinements to the thing queries.

Layer	PQ	PQ^{th}	PQ^{st}	Fps
1st	48.8	54.3	40.5	10.6
2nd	49.5	54.5	42.0	9.8
3rd	49.6	54.5	42.3	9.3
Last	49.6	54.4	42.4	7.8

Table 10. Results of each layer in the mask decoder.

Effect of Query Decoupling Strategy. We compare our proposed query decoupling strategy with previous DETR's matching method (described here as "joint matching") [1], as shown in Tab. 8. Following DETR, joint matching uses a set of queries to target both things and stuff and feeds all queries to both location decoder and mask decoder. For our proposed query decoupling strategy, we use thing queries to detect things through bipartite matching and use location decoder to refine them. Stuff queries are assigned through class-fixed assign strategy. For a fair comparison, both the joint matching strategy and our query decoupling strategy employ 353 queries. We can observe

Method	Clean	Mean	Blur				Noise				Digital				Weather			
			Motion	Defoc	Glass	Gauss	Gauss	Impul	Shot	Speck	Bright	Contr	Satur	JPEG	Snow	Spatt	Fog	Frost
Panoptic FCN (R50)	43.8	26.8	22.5	23.7	14.1	25.0	28.2	20.0	28.3	31.9	39.4	24.3	38.0	22.9	20.0	29.6	35.3	25.3
MaskFormer (R50)	47.0	29.5	24.9	28.1	16.4	29.5	31.2	24.7	30.9	34.8	42.5	27.5	41.2	22.0	20.4	31.0	38.5	27.7
D-DETR (R50)	47.6	30.3	25.6	28.7	16.8	29.7	32.5	24.9	31.4	35.9	43.1	28.6	41.3	24.5	21.7	31.7	39.7	28.7
Ours (R50)	50.0	32.9	26.9	30.2	17.5	31.6	35.5	27.9	35.4	38.6	45.7	31.3	43.9	29.0	24.3	35.0	41.9	32.3
MaskFormer (Swin-L)	52.9	41.7	37.3	38.0	30.4	39.3	42.3	42.5	42.8	45.3	49.7	43.9	49.4	39.7	35.2	45.2	48.8	37.9
Ours (Swin-L)	**55.8**	**47.2**	41.3	41.5	34.3	42.7	48.6	49.5	48.8	50.3	53.8	50.1	53.5	46.9	44.8	51.5	53.3	44.3
Ours (PVTv2-B5)	55.6	47.0	41.5	41.1	36.1	42.5	48.4	49.6	48.4	50.4	53.5	50.8	53.0	46.2	42.4	50.3	52.9	44.3

Table 11. Panoptic segmentation results on COCO-C. To ease the workload of the experiment, we use a subset of 2000 images from the COCO `val2017`. The third column is the average results on 16 types of corruption data.

that our proposed strategy highly boost PQ^{st}. In addition, panoptic segmentation model can perform instance segmentation by utilizing its thing results only. However, previous panoptic segmentation methods always perform poorly on instance segmentation task even though the two tasks are closely related. Tab. 8 shows both panoptic segmentation and instance segmentation performance of various methods. Our query decoupling strategy can achieve sota performance on panoptic segmentation task while obtaining a competitive instance segmentation performance. In short, query decoupling strategy achieves higher PQ^{st} and AP^{seg} compared to joint matching. We analyze the experimental results of joint matching and find that if one query prefers things more, the precision of stuff results detected by it will be lower, see Fig. 7. Each point represents the Thing-Preference and Stuff-Precision corresponding to each query, and the specific definitions are in Appendix. The red line is the linear regression of these points. When using one query set to detect things and stuff together, it will cause interference within each query. Our query decoupling strategy prevents things and stuff from interfering within the same query.

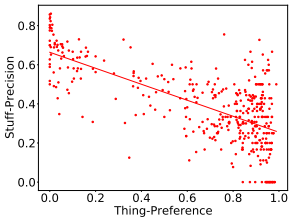


Figure 7. Things-Preference *vs.* Stuff-Precision.

4.4. Robustness to Natural Corruptions

Panoptic segmentation has promising applications in many fields, such as autonomous driving. Model robustness is one of the top concerns of autonomous driving. In this experiment, we evaluate the robustness of our model to disturbed data. We follow [42] and generate COCO-C, which extends the COCO validation set to include disturbed data generated by 16 algorithms from blur, noise, digital and weather categories. We compare our model to Panoptic FCN [21], D-DETR-MS and MaskFormer [3]. The results are shown in Tab. 11. We calculated the mean results of disturbed data on COCO-C. Using the same backbone, our model always performs better than others. Previous literature [43–45] found that transformer-based model has stronger robustness on image classification and semantic segmentation tasks. Our experimental results also show that the transformer-based backbone (Swin-L and PVTv2-B5) can bring better robustness to the model. However, for tasks requiring a more complex pipeline, such as panoptic segmentation, we argue that the design of the task head also plays an important role for the robustness of the model. For example, Panoptic SegFormer (Swin-L) has an average result of 47.2% PQ on COCO-C, outperforming MaskFormer (Swin-L) by 5.5% PQ, higher than their gap (2.9% PQ) on clean data. We posit it is due to our transformer-based mask decoder having stronger robustness than the convolution-based pixel decoder of MaskFormer.

5. Conclusion

Limitation. This work relies on deformable attention to process multi-scale features, and the speed is a little slow. Our model is still hard to handle features with a larger spatial shape and does not perform well for small targets.

Discussion. Recently, the segmentation field attempted to use a uniform pipeline to process various tasks, including semantic segmentation, instance segmentation, and panoptic segmentation. However, we think that complete unification is conceptually exciting but not necessarily a suitable strategy. Given the similarities and differences among the various segmentation tasks, "seek common ground while reserving differences" is a more reasonable guiding ideology. With query decoupling strategy, we can handle things and stuff in the same paradigm since they are represented as queries. In addition, we can also design customized pipelines for things or stuff. Such a flexible strategy is more suitable for various segmentation tasks. At present, task-specific designs still bring better performance. We encourage the community to further explore the unified segmentation frameworks and expect that Panoptic SegFormer can inspire future works.

6. Acknowledge

This work is supported by the Natural Science Foundation of China under Grant 61672273 and Grant 61832008. Ping Luo is supported by the General Research Fund of HK No.27208720 and 17212120. Wenhai Wang and Tong Lu are corresponding authors.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7

[2] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6

[3] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1, 2, 3, 5, 6, 7, 8

[4] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-Net: Towards unified image segmentation. In *NeurIPS*, 2021. 1, 2, 3, 5, 6

[5] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv:2106.13797*, 2021. 1, 6

[6] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1, 2, 4, 5

[7] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1, 2, 3, 6

[8] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 1

[9] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019. 1, 2

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1

[11] 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 *ECCV*, 2014. 2, 5

[12] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 2, 3, 4, 6, 7

[13] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6910–6919, October 2021. 2, 3, 6

[14] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 2, 6

[15] Ujwal Bonde, Pablo F Alcantarilla, and Stefan Leutenegger. Towards bounding-box free panoptic segmentation. In *DAGM German Conference on Pattern Recognition*, 2020. 2

[16] Qizhu Li, Xiaojuan Qi, and Philip HS Torr. Unifying training and inference for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13328, 2020.

[17] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 2

[18] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv:1902.05093*, 2019. 2

[19] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. SSAP: Single-shot instance segmentation with affinity pyramid. In *ICCV*, 2019. 2

[20] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, 2019. 2

[21] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *CVPR*, 2021. 2, 3, 6, 7, 8

[22] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 2, 4

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[25] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Solq: Segmenting objects by learning queries. *NeurIPS*, 2021. 3, 6

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015. 3

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 3, 5

[28] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. *NeurIPS*, 2020. 3, 4, 5, 6

[29] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 4

[30] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *CVPR*, 2016. 4

[31] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *International conference on 3D vision (3DV)*, 2016. 5

[33] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5, 6

[34] 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. *ICCV*, 2021. 6

[35] Chongsong Chen, Jiawei Ren, Daisheng Jin, Zhongang Cai, Cunjun Yu, Bairun Wang, Mingyuan Zhang, and Jinyi Wu. Joint coco and mapillary workshop at iccv 2019: Coco panoptic segmentation challenge track technical report: Panoptic htc with class-guided fusion. *SHR*, 56(84.1):67–2. 6

[36] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Yukang Chen, Lu Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation with point-based supervision. *arXiv preprint arXiv:2108.07682*, 2021. 6

[37] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9080–9089, 2020. 6

[38] Yangxin Wu, Gengwei Zhang, Hang Xu, Xiaodan Liang, and Liang Lin. Auto-panoptic: Cooperative multi-component architecture search for panoptic segmentation. *Advances in Neural Information Processing Systems*, 33, 2020. 6

[39] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 6

[40] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 6

[41] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15334–15342, 2021. 6, 7

[42] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8828–8838, 2020. 8

[43] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 8

[44] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*, 2021. 8

[45] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021. 8