electronics

MDPI

Article

Center-Guided Transformer for Panoptic Segmentation

Jong-Hyeon Baek [1], Hee Kyung Lee [2], Hyon-Gon Choo [2], Soon-heung Jung [2] and Yeong Jun Koh [1,*]

[1] Department of Computer Science & Engineering, Chungnam National University, Daejeon 34134, Republic of Korea; whdgusdl97@gmail.com
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; lhk95@etri.re.kr (H.L.); hyongonchoo@etri.re.kr (H.-G.C.); zeroone@etri.re.kr (S.-h.J.)
* Correspondence: yjkoh@cnu.ac.kr

Abstract: A panoptic segmentation network to predict masks and classes for things and stuff in images is proposed in this work. Recently, panoptic segmentation has been advanced through the combination of the query-based learning and end-to-end learning approaches. Current research focuses on learning queries without distinguishing between thing and stuff classes. We present decoupling query learning to generate effective thing and stuff queries for panoptic segmentation. For this purpose, we adopt different workflows for thing and stuff queries. We design center-guided query selection for thing queries, which focuses on the center regions of individual instances in images, while we set stuff queries as randomly initialized embeddings. Also, we apply a decoupling mask to the self-attention of query features to prevent interactions between things and stuff. In the query selection process, we generate a center heatmap that guides thing query selection. Experimental results demonstrate that the proposed panoptic segmentation network outperforms the state of the art on two panoptic segmentation datasets.

Keywords: panoptic segmentation; transformer; center-guided query selection

check for updates

Citation: Baek, J.-H.; Lee, H.K.; Choo, H.-G.; Jung, S.-h.; Koh, Y.J.; Center-Guided Transformer for Panoptic Segmentation. *Electronics* **2023**, *12*, 4801. https://doi.org/10.3390/electronics12234801

Academic Editors: Zhenhua Guo and Yue Wu

Received: 28 September 2023
Revised: 14 November 2023
Accepted: 15 November 2023
Published: 27 November 2023

1. Introduction

Panoptic segmentation [1] is the task in the domain of computer vision, involving the segmentation of both things and stuff. Things are defined as distinguishable and individual instances such as people, cars, and animals, where each instance contains unique id and class. In contrast, stuff means amorphous regions and encompassing areas such as the sky, meadows, grass, and other similar homogeneous areas. Starting from ConvNet-based panoptic segmentation models [2–5], recent panoptic segmentation methods [6–9] employ transformers to learn thing and stuff queries in various ways.

DETR [7] and its variants [8,9] set queries as randomly initialized embeddings and train the queries with a transformer decoder, as shown in Figure 1a. Then, the learned queries are transformed into class and mask predictions for things and stuff. Next, as in Figure 1b, the query selection approach, which chooses effective features from image features based on the class probability, is adopted in object detection methods [10,11]. However, things and stuff have different properties. Things are countable and contain small segments, while stuff is uncountable and includes large segments; thus, thing and stuff queries need to be learned differently. Also, the aforementioned approaches mix query features using self-attention in the transformer decoder, which yields interactions between thing and stuff queries.

In this work, we propose an architecture that integrates effective query selection for things and a decoupling mask to prevent things and stuff from interrupting each other, as illustrated in Figure 1c. First, we develop center-guided query selection for things, which exploits the center regions of instances from image features. To analyze center regions, we estimate a center heatmap, which has high values at the center of individual instances, to generate a center-guided feature. Based on the center-guided feature, we select the effective thing queries. After treating stuff queries as randomly initialized embeddings, we separately train thing and stuff queries using a transformer decoder with the decoupling mask. Using the trained thing and stuff queries, we obtain panoptic segmentation results from mask and class predictions. Experimental results demonstrate that the proposed panoptic segmentation network outperforms the state of the art on the COCO panoptic dataset [12] and ADE20K panoptic dataset [13]. Specifically, the proposed network yields the best performance for things, while it provides comparable results for stuff with respect to the state of the art on the COCO panoptic dataset. The proposed network achieves 52.2 PQ and 44.1 $\text{AP}^{\text{th}}_{\text{pan}}$ on the COCO panoptic dataset, and 41.5 PQ and 28.9 $\text{AP}^{\text{th}}_{\text{pan}}$ on the ADE20K panoptic dataset, where the metrics PQ and $\text{AP}^{\text{th}}_{\text{pan}}$ have a range from 0 to 100.

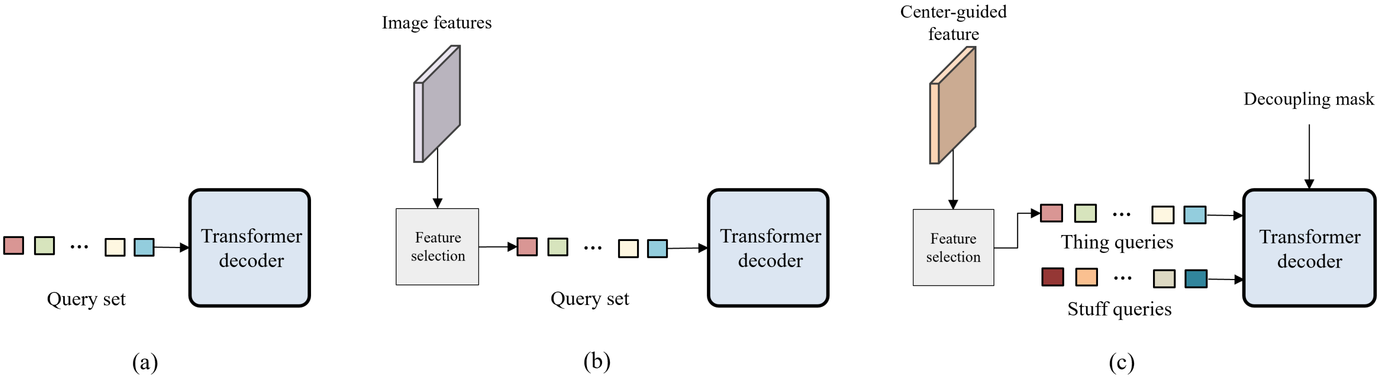

Figure 1. Approaches for query learning: (**a**) randomly initialized embeddings, (**b**) query selection, and (**c**) proposed center-guided query selection and decoupling mask.

The rest of this paper is organized as follows: Section 2 surveys panoptic segmentation methods and center-based learning techniques. Section 3 describes the proposed network. Section 4 discusses the experimental results. Finally, Section 5 concludes this work and provides future work directions.

2. Related Works

Panoptic segmentation [1] is a joint task including the semantic segmentation and instance segmentation tasks, requiring the prediction of distinct masks to represent both things and stuff. Early panoptic segmentation methods attempt to combine the existing semantic segmentation network and instance segmentation network effectively. For example, UPSNet [2] utilizes two separate branches to produce semantic and instance segmentation, and then it subsequently integrates both results using an additional panoptic segmentation head. PanopticFCN [3] jointly models things and stuff networks by designing a unified convolution pipeline to simplify panoptic segmentation.

Recently, transformer-based models [6–9] have achieved the promising performance in panoptic segmentation. DETR [7] is an end-to-end solution to address both object detection and panoptic segmentation tasks. However, it is still inferior to classical segmentation models, since DETR produces panoptic segmentation by adding a simple mask head on top of object detection networks. MaskFormer [8] and Mask2Former [9] have a architectures similar to that of DETR but differ in using a global segmentation decoder and some specialized designs for mask prediction. MaskFormer [8] builds a pixel decoder to generate mask predictions through simple matrix multiplication between enhanced queries and pixel decoder output. Mask2Former [9] proposes masked attention, which uses mask predictions in the process of self-attention, to reduce training time significantly. Panoptic Segformer [6] adopts an auxiliary location decoder to assist instance queries to learn location clues and ease model training. These transformer-based methods rely on learnable queries initialized with random values to estimate things and stuff. On the contrary, we propose a query selection algorithm to extract individual queries from image features using object center information. Also, fast panoptic segmentation networks [5,14,15] are presented. YOSO [15] developed the feature pyramid aggregator for speedup in GPU latency and the separable dynamic decoder for generating panoptic kernels. IDNet [14] decomposes panoptic segmentation into category and location information, which simplifies the network architecture.

The object's center is able to provide a rich context for solving various computer vision tasks, such as object detection [16–19] and segmentation [5,20]. CenterNet [16] detects each object as center keypoints. CenterNet2 [17] further enhances center representation using a heatmap approach. FCOS [18] introduces a centerness branch to predict the deviation of a pixel from the center of its corresponding box. ExtremeNet [19] predicts geometric centers and aligns them into a bounding box. In the segmentation task, CenterMask [20] leverages a center heatmap for anchor-free instance segmentation. Panoptic-DeepLab [5] first estimates all foreground masks from an image and then extracts thing classes based on instance centers. On the other hand, we take into account centers of object instances to extract instance queries from the corresponding feature space. In the experimental results, the proposed algorithm exhibits the superior performance in comparison to the other panoptic segmentation models.

3. Proposed Methods

3.1. Architecture

Figure 2 illustrates an overview of the proposed panoptic segmentation network. In this section, we introduce the proposed center-guided query selection module and transformer decoder with decoupling mask.

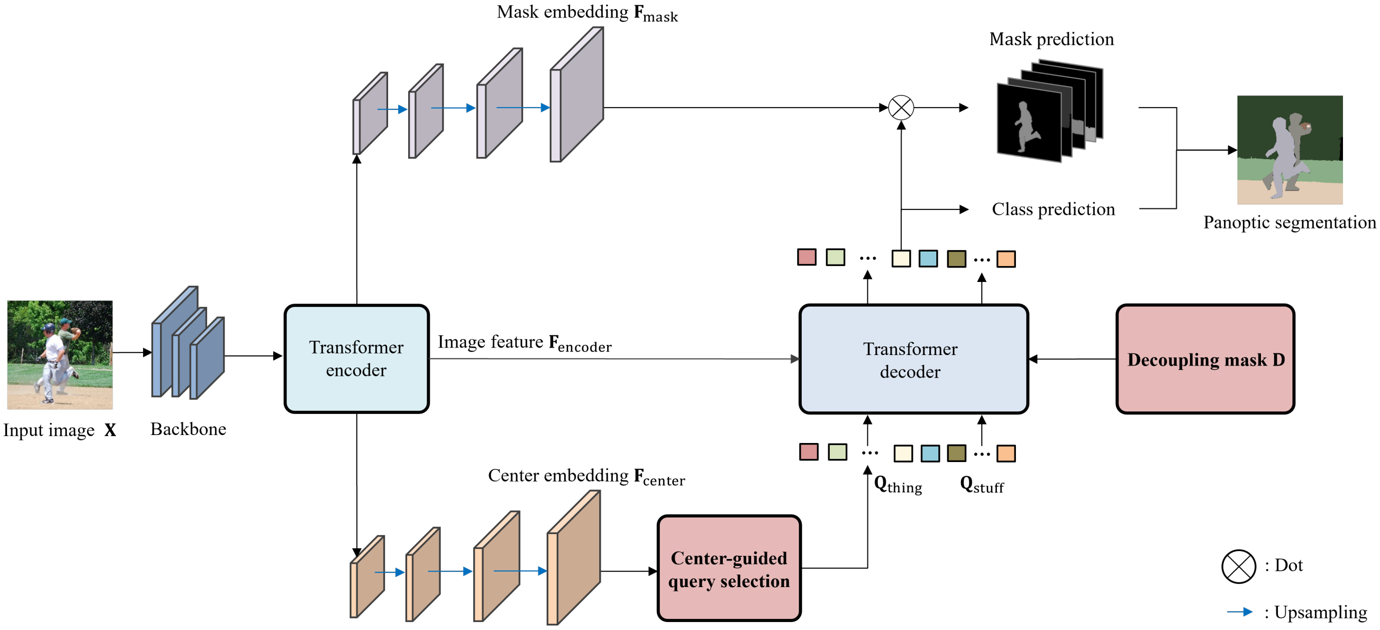
Mask embedding $\mathbf{F}_{\text{mask}}$
Mask prediction
Class prediction
Panoptic segmentation
Input image $\mathbf{X}$
Backbone
Transformer encoder
Image feature $\mathbf{F}_{\text{encoder}}$
Transformer decoder
Decoupling mask D
Center embedding $\mathbf{F}_{\text{center}}$
Center-guided query selection
$\mathbf{Q}_{\text{thing}}$
$\mathbf{Q}_{\text{stuff}}$
: Dot
: Upsampling

Figure 2. Overview of the proposed network. We use the backbone and transformer encoder to extract image feature $\mathbf{F}_{\text{encoder}}$ from an input image $\mathbf{X}$. $\mathbf{F}_{\text{encoder}}$ is gradually upsampled to obtain center embedding $\mathbf{F}_{\text{center}}$ (orange block) and mask embedding $\mathbf{F}_{\text{mask}}$ (purple block). Thing queries are selected from $\mathbf{F}_{\text{center}}$ through the center-guided query selection process, while stuff queries are randomly initialized. The transformer decoder generates enhanced queries based on the attention mechanism between $\mathbf{F}_{\text{encoder}}$ and queries with decoupling mask $\mathbf{D}$. Then, the enhanced queries are transformed into mask and class predictions for panoptic segmentation.

Backbone and transformer encoder: The backbone extracts image features from an input image $\mathbf{X} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, and the transformer encoder generates a new feature map $\mathbf{F}_{\text{encoder}} \in \mathbb{R}^{H_1 \times W_1 \times C}$ from the image features, where $H_1 = H_0/32$, $W_1 = W_0/32$, and $C = 2048$. We employ ResNet50 [21] for the backbone and the transformer encoder in [9]. The transformer encoder consists of deformable attention [10], layer normalization, and a feed forward network (FFN). Feature map $\mathbf{F}_{\text{encoder}}$ is gradually upsampled to a center embedding $\mathbf{F}_{\text{center}} \in \mathbb{R}^{H \times W \times C}$ and a mask embedding $\mathbf{F}_{\text{mask}} \in \mathbb{R}^{H \times W \times C}$ through the two sets of convolution layer and bilinear interpolation operation, where $H = 4H_1$, $W = 4W_1$. Also, $\mathbf{F}_{\text{encoder}}$ is fed into the transformer decoder for attention mechanisms with queries.

Center-guided query selection: Traditional transformer-based panoptic segmentation models [6,8,9] typically use randomly initialized embeddings to learn queries without distinguishing between things and stuff. The proposed network learns things and stuff separately to prevent thing queries and stuff queries from interrupting each other. Inspired by center-based learning for object detection [16–19], we develop the mechanism of center-guided query selection for thing queries. The center regions of individual instances in input images contain the cues to distinguish different instances. Thus, we estimate a center heatmap to guide effective thing query selection.

Figure 3 shows the diagram of the proposed center-guided query selection. Center embedding $\mathbf{F}_{\text{center}}$ passes through the FFN to estimate center heatmap $\mathbf{H} \in \mathbb{R}^{H \times W}$, which contains the location information of the instances. Then, we obtain center-guided feature $\tilde{\mathbf{F}}_{\text{center}} \in \mathbb{R}^{H \times W \times C}$ using element-wise multiplication between the estimated center heatmap $\mathbf{H}$ and each channel of $\mathbf{F}_{\text{center}}$. To this end, we employ the feature selection process in [10,11] to determine the top K query features from center-guided feature $\tilde{\mathbf{F}}_{\text{center}}$. $\tilde{\mathbf{F}}_{\text{center}}$ passes through a linear layer and softmax to obtain class probability map $\mathbf{P} \in \mathbb{R}^{H \times W \times C_{\text{thing}}}$ for things, where C_{thing} is the number of thing classes. Then, we pick the highest probability from $\mathbf{P}$ for each pixel and construct thing query $\mathbf{Q}_{\text{thing}} \in \mathbb{R}^{N_{\text{thing}} \times C}$, where $N_{\text{thing}} = K$, by selecting the top K features from $\tilde{\mathbf{F}}_{\text{center}}$ in terms of the highest probabilities extracted from $\mathbf{P}$. Since center heatmap $\mathbf{H}$ has high values on the central parts of the instances, $\mathbf{H}$ conveys strong visual patterns related to the instances to obtain effective thing query $\mathbf{Q}_{\text{thing}}$. Note that we only perform center-guided query selection for thing queries $\mathbf{Q}_{\text{thing}}$, while we simply set stuff queries $\mathbf{Q}_{\text{stuff}} \in \mathbb{R}^{N_{\text{stuff}} \times C}$ as N_{stuff} randomly initialized embeddings.

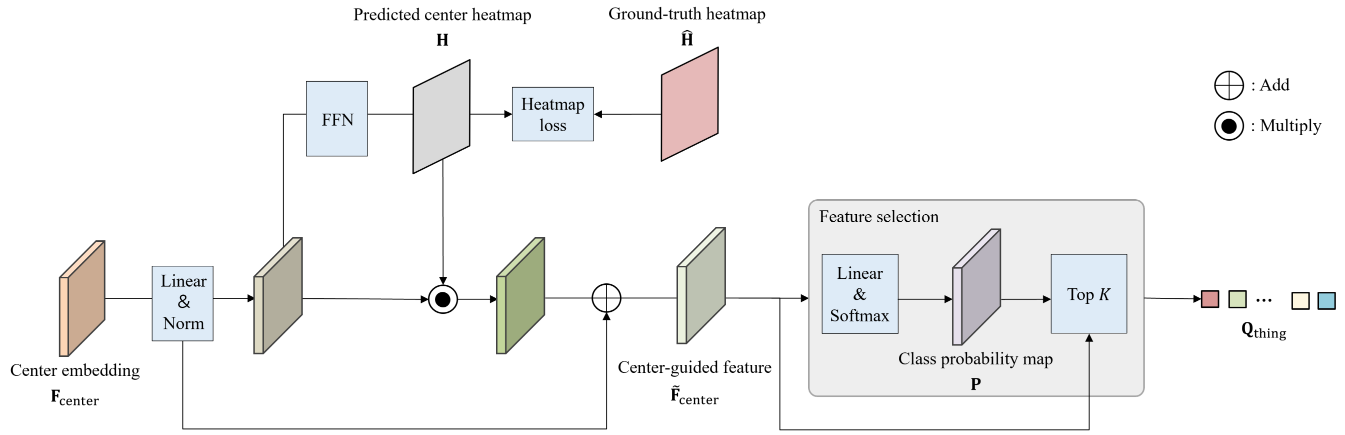

Figure 3. Diagram of center-guided query selection. It generates center-guided feature $\tilde{\mathbf{F}}_{\text{center}}$ using the estimated heatmap $\mathbf{H}$, which contains the center information of the instances. Feature selection extracts K queries from center-guided feature $\tilde{\mathbf{F}}_{\text{center}}$ based on thing class probabilities.

Transformer decoder with decoupling mask: We need to train queries to inject enough information to derive classes and masks. For this purpose, $\mathbf{Q}_{\text{thing}}$ and $\mathbf{Q}_{\text{stuff}}$ are concatenated as $\mathbf{Q} = [\mathbf{Q}_{\text{thing}}^T \; \mathbf{Q}_{\text{stuff}}^T]^T$ and fed to the transformer decoder, which includes self-attention, deformable attention, and the FFN, as in Figure 4. Considering the different properties between things and stuff, we apply decoupling mask $\mathbf{D} \in \mathbb{R}^{(N_{\text{thing}}+N_{\text{stuff}}) \times (N_{\text{thing}}+N_{\text{stuff}})}$ to self-attention, where $\mathbf{D}$'s element $\mathbf{D}(i,j)$ is defined as

$$\mathbf{D}(i,j) = \begin{cases} 0 & \text{if } i,j \leqq N_{\text{thing}}, \\ 0 & \text{if } N_{\text{thing}} < i,j \leqq N_{\text{thing}} + N_{\text{stuff}}, \\ -\infty & \text{otherwise.} \end{cases} \quad (1)$$

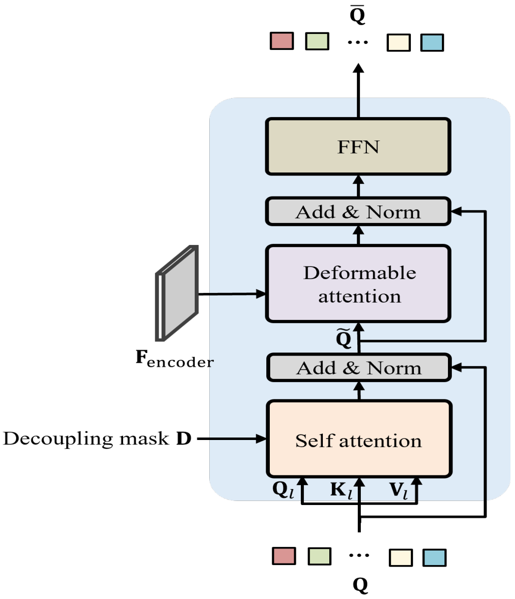

Figure 4. Diagram of the transformer decoder. Given query $\mathbf{Q}$, it generates enhanced query $\bar{\mathbf{Q}}$ through self-attention and deformable attention with image feature $\mathbf{F}_{\text{encoder}}$.

Then, self-attention in the transformer decoder is formulated as

$$\tilde{\mathbf{Q}} = \text{Layernorm}(\text{softmax}(\mathbf{D} + \mathbf{Q}_l\mathbf{K}_l^T)\mathbf{V}_l + \mathbf{Q}) \tag{2}$$

where $\mathbf{Q}_l$, $\mathbf{K}_l$, and $\mathbf{V}_l$ are the query, key, and value extracted from $\mathbf{Q}$ through a linear layer, respectively. We prevent interference between thing and stuff queries using decoupling mask $\mathbf{D}$. For the stability of the learning process, we use the residual connection with $\mathbf{Q}$ and perform layer normalization after the residual connection. After the self-attention process, we use deformable attention to inject $\mathbf{F}_{\text{encoder}}$ into $\tilde{\mathbf{Q}}$, resulting in enhanced query set $\bar{\mathbf{Q}}$.

Estimation: Masks and classes are estimated from enhanced query set $\bar{\mathbf{Q}}$. First, masks are computed using the dot product between $\bar{\mathbf{Q}}$ and mask embedding $\mathbf{F}_{\text{mask}}$. Second, $\bar{\mathbf{Q}}$ passes through a fully connected layer to predict the class probability. Finally, we obtain panoptic segmentation results from mask and class predictions.

3.2. Loss

The proposed network outputs $N_{\text{thing}} + N_{\text{stuff}}$ predictions, including masks and classes. Then, we perform the Hungarian algorithm [22] to match predictions and ground truths, following [6–9]. For each match, we compute the focal loss [23] between class probability prediction $\mathbf{c}_k$ and ground truth $\hat{\mathbf{c}}_k$ as follows:

$$\mathcal{L}_{\text{c}}(\mathbf{c}_k, \hat{\mathbf{c}}_k) = \lambda_{\text{class}}[\{\alpha(1-\hat{\mathbf{c}}_k)^{\gamma} \cdot -\log(\hat{\mathbf{c}}_k) \cdot \mathbf{c}_k\} - \{(1-\alpha) \cdot \hat{\mathbf{c}}_k^{\gamma} \cdot -\log(1-\hat{\mathbf{c}}_k) \cdot (1-\mathbf{c}_k)\}] \tag{3}$$

where λ_{class}, α, and γ were experimentally set to 4, 0.25, and 2, respectively. Also, to compare the estimated mask $\mathbf{M}_k \in \mathbb{R}^{H_0 \times W_0}$ and ground truth $\hat{\mathbf{M}}_k$, we employ the mask loss ($\mathcal{L}_{\text{m}}(\mathbf{M}_k, \hat{\mathbf{M}}_k)$) in [8], which is composed of per-pixel cross-entropy loss $\mathcal{L}_{\text{pixel}}(\mathbf{M}_k, \hat{\mathbf{M}}_k)$ and dice loss [24] $\mathcal{L}_{\text{dice}}(\mathbf{M}_k, \hat{\mathbf{M}}_k)$:

$$\mathcal{L}_{\text{m}}(\mathbf{M}_k, \hat{\mathbf{M}}_k) = \lambda_{\text{pixel}}\mathcal{L}_{\text{pixel}}(\mathbf{M}_k, \hat{\mathbf{M}}_k) + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}(\mathbf{M}_k, \hat{\mathbf{M}}_k) \tag{4}$$

where λ_{pixel} and λ_{dice} were set to 5 and 5, according to [9]. Additionally, to train the center-guided query selection module, we generate ground-truth heatmap $\hat{\mathbf{H}}$ by applying Gaussian distributions to all instance center points for each image. Then, we compute the focal loss between the predicted center heatmap $\mathbf{H}$ and $\hat{\mathbf{H}}$.

4. Experiments

4.1. Setting

Dataset: We conducted experiments on the proposed network on two panoptic segmentation datasets: the COCO panoptic [12] and ADE20K panoptic [13] datasets. The COCO panoptic dataset consists of annotated images with mask and class labels for 80 thing and 53 stuff classes. The COCO panoptic dataset is divided into training set, validation set, and test set, which contain 118,785, 5000, and 5000 images, respectively. The ADE20K panoptic dataset provides object- and semantic-level information for object detection and segmentation. It consists of 100 thing classes and 50 stuff classes. The dataset contains 20,210 images for the training set and 2000 images for the validation set. Figures 5 and 6 show examples of the COCO panoptic and ADE20K panoptic datasets.

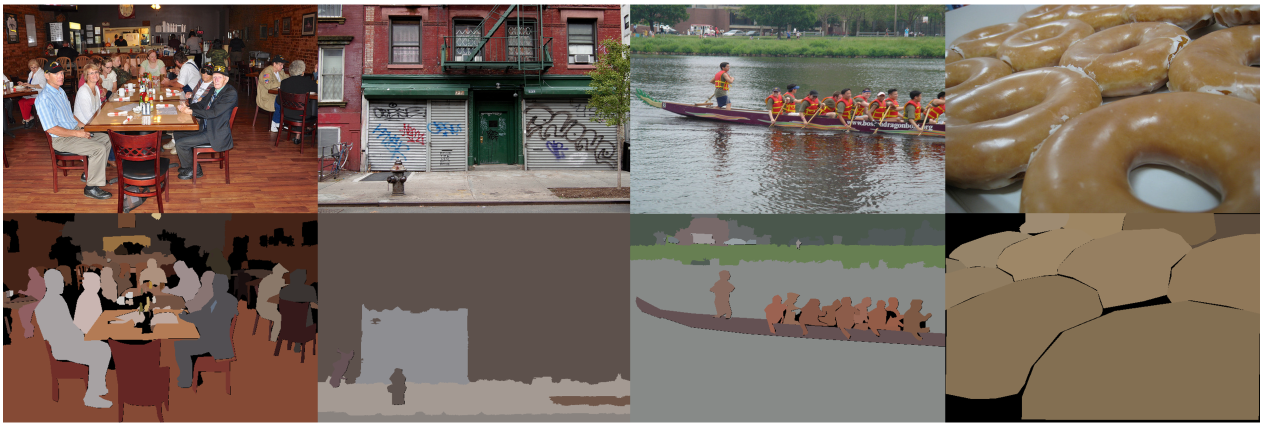

Figure 5. Examples of COCO panoptic images [12]. The first row and the second row represent images and their annotations, respectively.

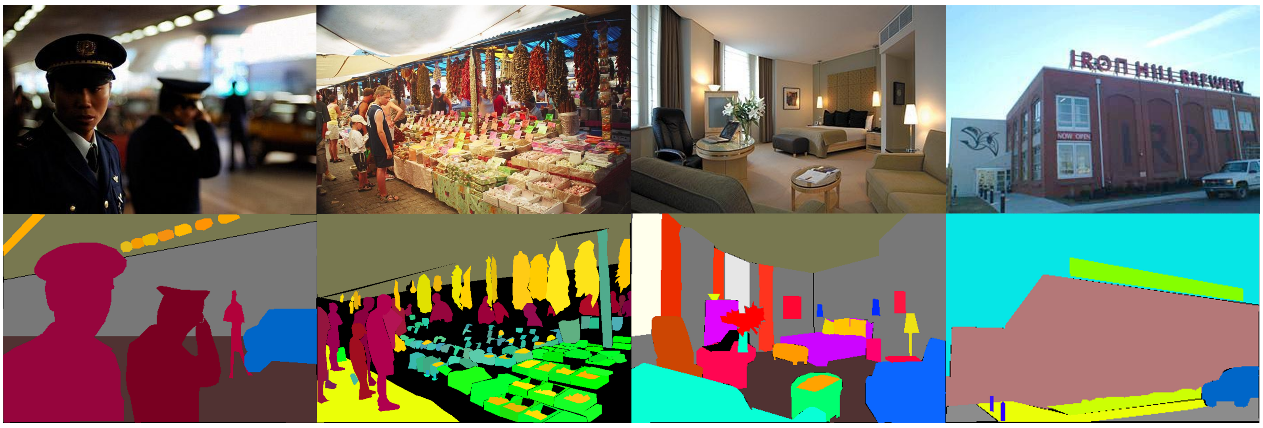

Figure 6. Examples of ADE20K panoptic images [13]. The first row and the second row represent images and their annotations, respectively.

Implementation details and training settings: We implemented the model using the detectron2 [25] platform, based on PyTorch. For training, the size of the input images was set to 1280 × 1280 on COCO, while it was set to 640 × 640 on ADE20K. We employed the standard convolution-based ResNet50 [21], ResNet101 [21] and Swin-Transformer [26] as the backbone network. The transformer encoder and the transformer decoder were repeated six times and nine times, respectively. The numbers of thing and stuff queries were 300 and 53, respectively. During the training process, we used four NVIDIA RTX A6000 GPUs, with a batch size of 4 per GPU. For training the proposed network, we set epoch to 50 for the COCO panoptic dataset and epoch to 120 for the ADE20K panoptic dataset. We optimized the proposed network using the AdamW optimizer. The initial learning rate was set to 1×10^{-4}, and the multiple step learning rate scheduling technique was applied to decay the learning rate at specific epochs. The reduction rate is set to 1/10, and the learning rate gradually decreases at 36 epoch and 48 epoch for the COCO panoptic dataset, while it decreases at 75 epoch and 105 epoch for the ADE20K panoptic dataset.. The weight decay value is set to 0.05.

Evaluation metrics: For evaluation, we used Panoptic Quality (PQ) [1] to measure the performance in both classification and segmentation. Also, we used the additional metric of $\text{AP}^{\text{th}}_{\text{pan}}$, which measures the average precision (AP) of segmentation for thing categories to demonstrate the effectiveness of the proposed method for thing classes. Both PQ and $\text{AP}^{\text{th}}_{\text{pan}}$ had a range from 0 to 100.

4.2. Comparison with Other Methods

COCO panoptic dataset: In Table 1, we compare the proposed method with existing panoptic segmentation methods [3,5–9,14,27] on the COCO panoptic [12] dataset. Table 1 shows the PQ, $\text{AP}^{\text{th}}_{\text{pan}}$ scores of the existing methods, which were obtained from the respective papers. We see that the proposed method outperforms both non-transformer-based methods [3,5] and transformer-based ones [6–9]. Specifically, the proposed method surpasses the prior state of the art (Mask2Former [9]) by margins of 0.3 and 2.4 in terms of PQ and $\text{AP}^{\text{th}}_{\text{pan}}$, respectively. The proposed method achieves the remarkable performance for thing classes (58.4 PQ_{thing}), which indicates that the proposed center-guided query selection is essential to exploiting features for different instances in each image. The highest score of $\text{AP}^{\text{th}}_{\text{pan}}$ shows that the proposed network is effective in the segmentation of thing classes. Figure 7 shows the qualitative comparison of the proposed method with MaskFormer and Mask2Former on the COCO panoptic dataset.

As shown in Figure 7, the proposed network provides more accurate segmentation results and distinguishes different instances compared with MaskFormer and Mask2Former. For example, in the third row in Figure 7, the proposed method significantly enhances the detection performance of things, resulting in more segmentation thing masks than both MaskFormer and Mask2Former. Specifically, while MaskFormer merges the several masks of individual cakes into a single mask, Mask2Former completely fails to detect and segment the cake instances. On the other hand, the proposed method faithfully detect individual cake instances and provides accurate segmentation mask results. Moreover, as illustrated in the stuff region in the fourth row, MaskFormer incorrectly classifies the fence class into the tree class and yields merged masks. Also, Mask2Former fails to obtain segmentation masks for tree regions. In contrast, the proposed method provides remarkable mask results for stuff classes, including tree, fence, window, and wall-brick.

Table 1. Comparison of the proposed method with existing panoptic segmentation networks on the COCO panoptic [12] val2017 dataset. The best results are boldfaced.

Model	**Backbone**	**PQ**	**PQ_{thing}**	**PQ_{stuff}**	**$\text{AP}^{\text{th}}_{\text{pan}}$**	**FLOPs**	**Params**
Panoptic DeepLab [5]	Xception71 [28]	41.4	45.1	35.9	-	-	-
ChaInNet [27]	ResNet50	43.0	49.8	33.8	-	-	-
DETR [7]	ResNet50	43.2	48.2	36.1	31.1	248 G	43 M
Panoptic FCN [3]	ResNet50	43.6	49.3	35.0	36.6	244 G	37 M
IDNet [14]	ResNet50	43.8	49.6	35.0	-	-	-
MaskFormer [8]	ResNet50	46.5	51.0	39.8	33.0	181 G	45 M
Panoptic Segformer [6]	ResNet50	49.6	54.4	42.4	39.5	214 G	51 M
Mask2Former [9]	ResNet50	51.9	57.7	**43.0**	41.7	226 G	44 M
Ours	ResNet50	**52.2**	**58.4**	42.6	**44.1**	276 G	51 M

ADE20K panoptic dataset: Table 2 compares the proposed method with IDNet [14], MaskFormer [8], Panoptic Segformer [6], YOSO [15], and Mask2Former [9] in terms of PQ and $\text{AP}^{\text{th}}_{\text{pan}}$ on ADE20K. The proposed method achieves the best performance in all metrics. Our method surpasses Mask2Former by over 1.8 and 2.7 in terms of PQ and $\text{AP}^{\text{th}}_{\text{pan}}$. Figure 8 shows the qualitative comparison of the proposed method with Mask-

Former and Mask2Former. We observe that the proposed method effectively distinguishes thing and stuff classes and yields more accurate mask and class results than MaskFormer and Mask2Former. For instance, in the fourth row in the Figure 8, MaskFormer produces inaccurate mask results for the chair class, leading to incorrect or incomplete masks for some chair instances. Mask2Former completely misses the chair and table instances. However, the proposed method yields accurate chair and table segmentation results. Furthermore, in the last row, MaskFormer fails to find wall regions, and thus it misclassifies wall regions into building or water classes. Mask2Former accurately predicts the stuff area, but fails to achieve accurate instance segmentation such as houses and stairs instances. In contrast, the proposed method not only accurately predicts segmentation results for stuff, but also precisely extracts thing segmentation results from the image.

Table 2. Comparison of the proposed method with existing panoptic segmentation networks on the ADE20K panoptic [13] validation dataset. The best results are boldfaced.

Model	**Backbone**	**PQ**	$\mathbf{PQ_{thing}}$	$\mathbf{PQ_{stuff}}$	$\mathbf{AP^{th}_{pan}}$
IDNet [14]	ResNet50	30.2	33.2	24.3	-
MaskFormer [8]	ResNet50	34.7	32.2	39.7	-
Panoptic Segformer [6]	ResNet50	36.4	35.3	38.6	-
YOSO [15]	ResNet50	38.0	37.3	39.4	-
Mask2Former [9]	ResNet50	39.7	38.8	40.5	26.2
Ours	ResNet50	**41.5**	**41.1**	**42.2**	**28.9**

4.3. Ablation Study

In Tables 3 and 4, we conduct ablation studies to validate the effectiveness of center-guided query selection and the decoupling mask. Tables 3 and 4 list the performance of the proposed network without center-guided query selection and the decoupling mask on COCO and ADE20K, respectively. As shown in Tables 3 and 4, both components improve the performance in all metrics. Specifically, center-guided query selection increases the PQ_{thing} and AP^{th}_{pan} scores by 1.2 and 2.1 on COCO, while it improves the PQ_{thing} and AP^{th}_{pan} scores by 1.7 and 2.8 on ADE20K. This indicates that center-guided feature $\tilde{\mathbf{F}}_{center}$ is effective in segmenting objects and distinguishing different instances. Also, without the decoupling mask, PQ_{thing} and PQ_{stuff} performance is degraded on both COCO and ADE20K. When we remove the two components, PQ scores are reduced by 1.8 and 2.1 on COCO and ADE20K, respectively. These results indicate that the proposed modules are essential for accurate panoptic segmentation.

For the learning of thing and stuff queries, there are three approaches: (1) center-guided query selection, (2) feature selection [10,11], and (3) random initialization. Table 5 shows an ablation study according to combinations of thing and stuff query learning. We observe that the combination of center-guided query selection for things and random initialization for stuff, i.e., the proposed method, yields the best performance. When feature selection is adopted for stuff instead of random initialization, we experience accuracy degradation. Also, the proposed combination surpasses the traditional feature selection in [10,11]. Table 6 lists the panoptic segmentation performance for various backbones: (1) ResNet50, (2) ResNet101, and (3) Swin-T [26]. By comparing ResNet50 and ResNet101, the performance is improved as parameters increase. Also, the transformer-based backbone [26] yields the best performance, even though it uses fewer parameters than ResNet101.

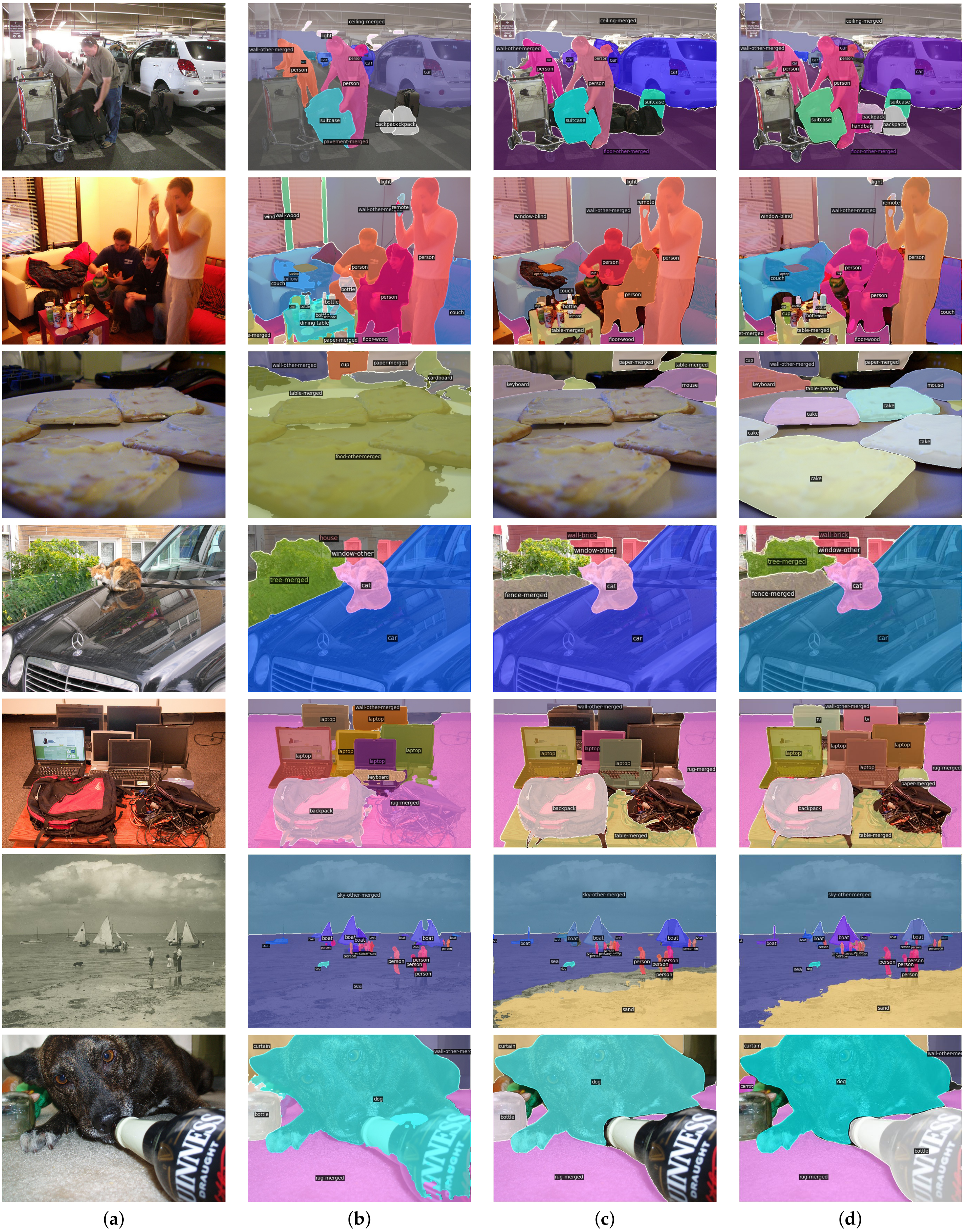

Figure 7. Qualitative comparison on the COCO panoptic [12] val2017 dataset. (**a**) Input; (**b**) MaskFormer; (**c**) Mask2Former; (**d**) ours.

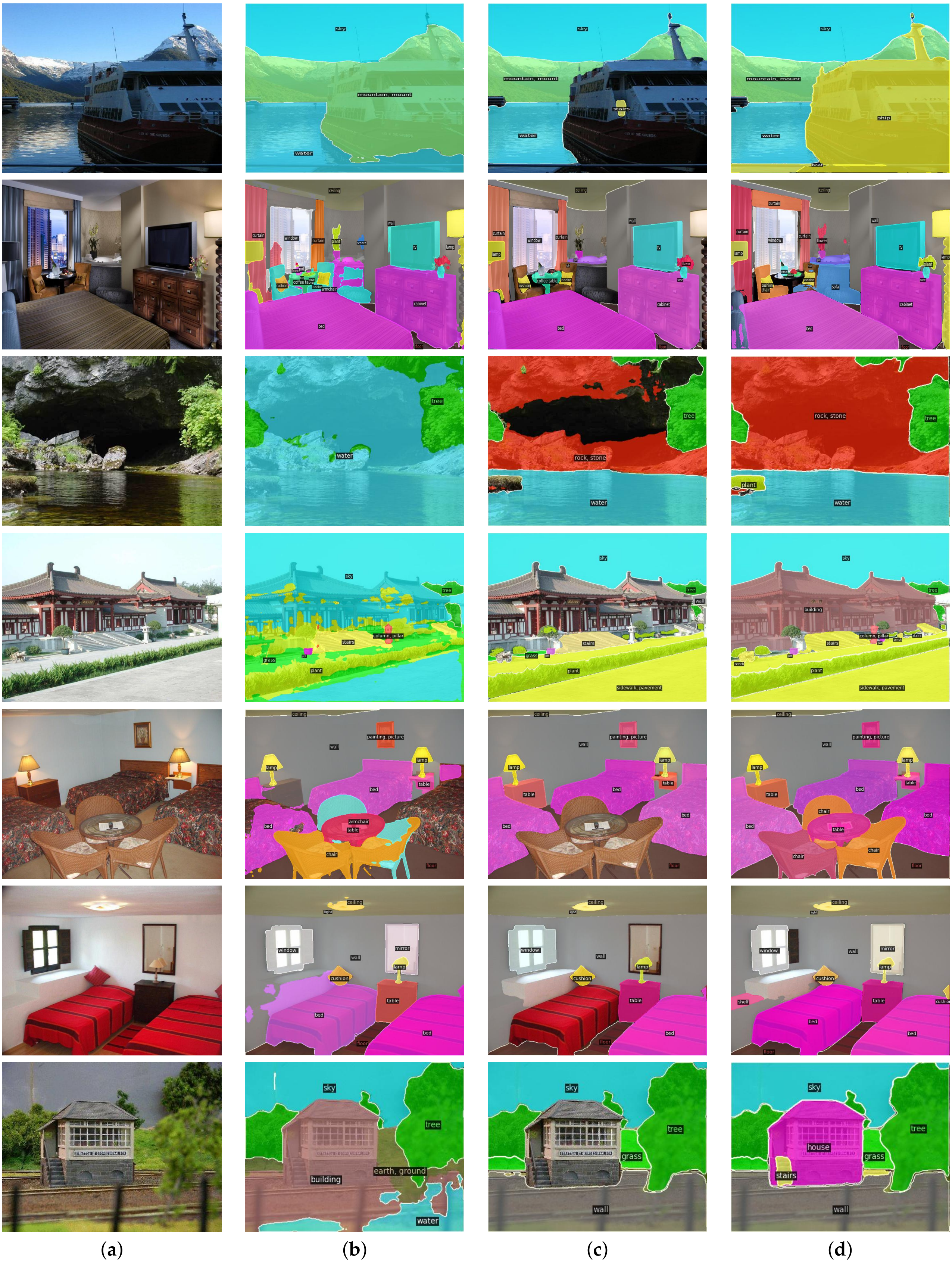

Figure 8. Qualitative comparison on the ADE20K panoptic [13] dataset. (**a**) Input; (**b**) MaskFormer; (**c**) Mask2Former; (**d**) ours.

Table 3. Ablation study on the COCO panoptic val2017 dataset according to different settings. The best results are boldfaced.

Model	**PQ**	**$\mathrm{PQ}_{\mathrm{thing}}$**	**$\mathrm{PQ}_{\mathrm{stuff}}$**	**$\mathrm{AP}^{\mathrm{th}}_{\mathrm{pan}}$**	**FLOPs**	**Params**
Ours	**52.2**	**58.4**	**42.6**	**44.1**	276 G	51 M
−Center-guided query selection	51.6	57.2	42.3	42.0	273 G	50 M
−Decoupling mask	51.8	57.6	42.1	42.6	273 G	50 M
−2 components above	50.4	56.2	41.3	41.4	270 G	50 M

Table 4. Ablation study on the ADE20K panoptic validation set according to different settings. The best results are boldfaced.

Model	**PQ**	**$\mathrm{PQ}_{\mathrm{thing}}$**	**$\mathrm{PQ}_{\mathrm{stuff}}$**	**$\mathrm{AP}^{\mathrm{th}}_{\mathrm{pan}}$**
Ours	**41.5**	**41.1**	**42.2**	**28.9**
−Center-guided query selection	40.2	39.4	41.1	26.1
−Decoupling mask	40.5	39.9	41.3	26.9
−2 components above	39.4	39.3	40.1	25.2

Table 5. Ablation study on the COCO panoptic val2017 dataset according to query learning settings. The best results are boldfaced.

	Center-Guided Query Selection	**Feature Selection**	**Random Initialization**	**PQ**	**$\mathrm{PQ}_{\mathrm{thing}}$**	**$\mathrm{PQ}_{\mathrm{stuff}}$**	**$\mathrm{AP}^{\mathrm{th}}_{\mathrm{pan}}$**
Things	✓			**52.2**	**58.4**	**42.6**	**44.1**
Stuff			✓				
Things	✓			51.8	58.2	42.1	43.6
Stuff		✓					
Things		✓		51.6	57.6	42.3	42.9
Stuff		✓					

Table 6. Ablation study on the COCO panoptic val2017 dataset with various backbones, ResNet50, ResNet101, and Swin-T [26]. The best results are boldfaced.

Backbone	**PQ**	**$\mathrm{PQ}_{\mathrm{thing}}$**	**$\mathrm{PQ}_{\mathrm{stuff}}$**	**$\mathrm{AP}^{\mathrm{th}}_{\mathrm{pan}}$**	**FLOPs**	**Params**
ResNet50	52.2	58.4	42.6	44.1	276G	51M
ResNet101	52.7	58.9	43.3	44.7	342G	69M
Swin-T	**53.6**	**59.8**	**43.6**	**45.2**	280G	48M

5. Conclusions

We propose a panoptic segmentation network to predict masks and classes for things and stuff. The key insight of the proposed network is to generate effective thing and stuff queries for panoptic segmentation. First, we developed center-guided query selection, which exploits center information for detecting and segmenting individual instances. Second, we applied a decoupling mask to the transformer decoder, which prevents the interaction between thing and stuff queries. Experiments on COCO and ADE20K validated that the proposed panoptic segmentation network outperforms the existing methods, especially with respect to things. Despite its effectiveness, the proposed panoptic segmentation network has a limitation with respect to stuff, as reported in Table 1. Therefore, it remains a future work direction to generate effective queries for stuff classes.

Author Contributions: Conceptualization, H.L., H.-G.C., and S.-h.J.; data curation, H.L. and S.-h.J.; formal analysis, J.-H.B.; methodology, Y.J.K. and J.-H.B.; resources, J.-H.B; software, J.-H.B.; writing—original draft, J.-H.B.; writing—review and editing, Y.J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by an Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-00011, Video Coding for Machine) and an Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2022-0-01200, Convergence security core talent training business (Chungnam National University)).

Data Availability Statement: The datasets used in this study are the COCO panoptic dataset and the ADE20K dataset. The COCO dataset is available at https://cocodataset.org/#home (accessed on 12 September 2014),and the ADE20K dataset is available at http://sceneparsing.csail.mit.edu/ (accessed on 16 June 2019).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9404–9413.
2. Xiong, Y.; Liao, R.; Zhao, H.; Hu, R.; Bai, M.; Yumer, E.; Urtasun, R. Upsnet: A unified panoptic segmentation network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8818–8826.
3. Li, Y.; Zhao H.; Qi, X.; Wang, L.; Li, Z.; Sun, J.; Jia, J. Fully convolutional networks for panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 214–223.
4. Li, Y.; Chen, X.; Zhu, Z.; Xie, L.; Huang, G.; Du, D.; Wang, X. Attention-guided unified network for panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7026–7035.
5. Cheng, B.; Collins, M. D.; Zhu, Y.; Liu, T.; Huang T. S.; Adam, H.; Chen, L.C. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 12475–12485.
6. Li, Z.; Wang, W.; Xie, E.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; Lu, T. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 1280–1289.
7. Nicolas C.; Francisco M.; Gabriel S.; Nicolas U.; Alexander K.; Sergey Z. End-to end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Virtual, 23–28 August 2020; pp 213–229.
8. Bowen C.; Alexander G.; Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 7–10 December 2021.
9. Bowen C.; Misra, I.; Schwing, A.G.; Kirillov, A.; Girdhar, R. Masked-attention mask transformer for universal image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 1290–1299.
10. Zhu, W.; Xizhou S. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
11. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
12. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12. September 2014; pp. 740–755.
13. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ADE20k dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 633–641.
14. Lin, G.; Li, S.; Chen, Y.; Li, X. IDNet: Information Decomposition Network for Fast Panoptic Segmentation. *IEEE Trans. Image Process.* **2023**, early access. [CrossRef] [PubMed]
15. Hu, J.; Huang, L.; Ren, T.; Zhang, S.; Ji, R.; Cao, L. You Only Segment Once: Towards Real-Time Panoptic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 17819–17829.
16. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 6569–6578.
17. Zhou, X.; Koltun, V.; Krähenbühl, P. Probabilistic two-stage detection. *arXiv* **2021**, arXiv:2103.07461. Available online: https://arxiv.org/abs/2103.07461 (accessed on 12 Mar 2021).
18. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA 16–20 June 2019; pp. 9627–9636.
19. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 850–859.

20. Lee, Y.; Park, J. Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 13906–13915.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
22. Stewart, R.J.; Andriluka, M.; Ng, A.Y. End-to-end people detection in crowded scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2325–2333.
23. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2980–2988.
24. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D vision, Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
25. Yuxin W.; Alexander K.; Francisco M.; Wan-Yen L.; Ross G. Detectron2. Available online: https://github.com/facebookresearch/detectron2 (accessed on 8 September 2023).
26. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE International Conference on Computer Vision, Virtual, 11–17 Octover 2021; pp. 10012–10022.
27. Mao, L.; Ren, F.; Yang, D.; Zhang, R. ChaInNet: Deep Chain Instance Segmentation Network for Panoptic Segmentation. *Neural Process. Lett.* **2023**, *55*, 615–630. [CrossRef]
28. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.