

# Panoptic, Instance and Semantic Relations: A Relational Context Encoder to Enhance Panoptic Segmentation

Shubhankar Borse\*   Hyojin Park\*   Hong Cai   Debasmit Das   Risheek Garrepalli  
Fatih Porikli  
Qualcomm AI Research<sup>†</sup>

{sborse, hyojinp, hongcai, debadas, rgarrepa, fporikli}@qti.qualcomm.com

## Abstract

This paper presents a novel framework to integrate both semantic and instance contexts for panoptic segmentation. In existing works, it is common to use a shared backbone to extract features for both things (countable classes such as vehicles) and stuff (uncountable classes such as roads). This, however, fails to capture the rich relations among them, which can be utilized to enhance visual understanding and segmentation performance. To address this shortcoming, we propose a novel Panoptic, Instance, and Semantic Relations (PISR) module to exploit such contexts. First, we generate panoptic encodings to summarize key features of the semantic classes and predicted instances. A Panoptic Relational Attention (PRA) module is then applied to the encodings and the global feature map from the backbone. It produces a feature map that captures 1) the relations across semantic classes and instances and 2) the relations between these panoptic categories and spatial features. PISR also automatically learns to focus on the more important instances, making it robust to the number of instances used in the relational attention module. Moreover, PISR is a general module that can be applied to any existing panoptic segmentation architecture. Through extensive evaluations on panoptic segmentation benchmarks like Cityscapes, COCO, and ADE20K, we show that PISR attains considerable improvements over existing approaches.

## 1. Introduction

Panoptic segmentation [22] provides a unifying framework encompassing both semantic and instance segmentation. Its objective is to segment an image into *things* and *stuff*. Things include countable objects, such as cars and pedestrians, and stuff refers to uncountable concepts like sky and vegetation. Generating individual masks for *things* is similar to instance segmentation, while predicting masks

\* These authors contributed equally.

<sup>†</sup>Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

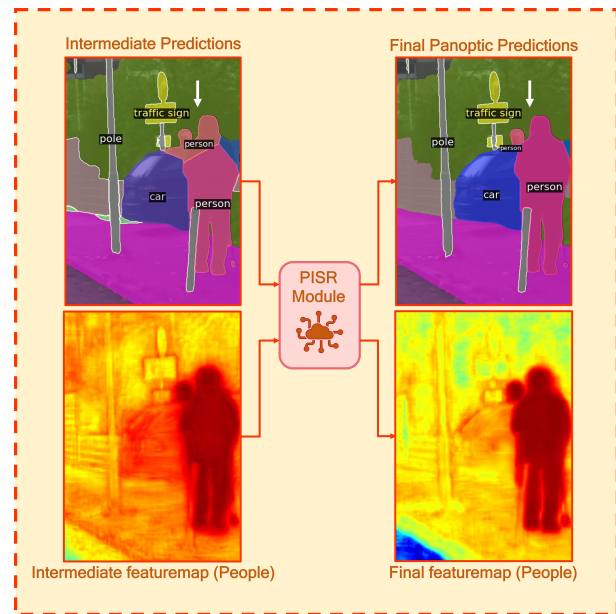


Figure 1. Our proposed PISR module takes features and predictions for semantic and instance segmentation as inputs and applies a relational attention scheme. In this image, we show the input/output pairs obtained by applying PISR to a Panoptic-DeepLab model with ResNet50 backbone. The improvements in both the features and panoptic predictions are clearly visible. PISR produces more confident and accurate predictions in regions where a lot of instances are cluttered together (shown with white arrows). The feature maps are visualized for the persons class.

for *stuff* is equivalent to performing semantic segmentation.

Since it aims to perform both tasks simultaneously, panoptic segmentation presents a challenge beyond conventional semantic or instance segmentation. Early works [22] propose to use separate modules for the two tasks, e.g., a Mask-RCNN module for instance segmentation and an FCN-based module for semantic segmentation. These two outputs are then combined during post-processing to generate panoptic segmentation. However, the accuracy in this case heavily relies on object detection quality. Besides, having two separate modules leads to redundant computations.

Recently, [27] proposed an architecture to process things and stuff together via a bottom-up, box-free approach. More specifically, semantic segmentation labels are first predicted, and then the instances are identified based on the grouped pixels. This architecture provides a unified approach, but it does not consider the relations among the semantic classes and instances. The object-contextual representations (OCR) module proposed by [51] allows modeling the relations across semantic classes. However, it is designed for the semantic segmentation task and does not take instance information into account, which is critical for panoptic segmentation. For example, two images can have the same semantic classes, but their respective instances in the images can be very different. Our intuition is that in each semantic class, some instances can have drastically different visual appearances, sizes, and poses, while others may look similar. Therefore, understanding the relations (e.g., visual similarities) among the classes and instances will benefit panoptic segmentation.

In this paper, we propose a novel *Panoptic, Instance, and Semantic Relations (PISR)* module, which captures the key relations among semantic classes and instances for panoptic segmentation. Given an image, PISR computes encodings for semantic classes and pivotal instances. During this process, it automatically identifies on which instances it should focus more. PISR then applies attention to these encodings as well as global features to capture rich contextual relations that are useful for the final panoptic segmentation. PISR is a general component that can be used with any panoptic segmentation network, such as Panoptic-DeepLab [11], Panoptic FPN [21], and Maskformer [12]. To the best of our knowledge, this is the first time that relations among both semantic classes and object instances are exploited for panoptic segmentation explicitly.

Our main contributions are summarized as follows:

- We propose a novel Panoptic, Instance, and Semantic Relations (PISR) model to capture the relations among semantic classes and instances, providing richer contexts that enhance panoptic segmentation performance.
- We devise a learnable scheme for PISR to automatically focus on more important instances while generating relational features. This provides a robust way to process the varying number of instances in each image.
- PISR is a universal module that can be used in any panoptic segmentation network. We show that it considerably enhances existing architectures on multiple datasets. We further conduct extensive ablation studies analyzing various aspects of PISR.

## 2. Related Work

**Panoptic Segmentation:** Aiming at assigning both instance and semantic labels to each image pixel, panoptic

segmentation combines instance and semantic segmentation.<sup>1</sup> One of the earliest methods for panoptic segmentation [22] utilizes predictions from separate instance and semantic segmentation models, but using it is inefficient as the two models do not share parameters. Recent approaches can be grouped into two categories: 1) separate representations and 2) unified representation. In the first case, instances and semantic classes are segmented using a single model, but through two different branches. Things are segmented either via boxes [10, 25, 50] or by using box-free methods [49]. Stuff is usually segmented using a fully convolutional branch. Other methods that use separate representations include AUNet [26], Panoptic FPN [21], and UPSNet [48]. In contrast, unified representation approaches segment things and stuff based on features generated from shared layers [24, 27]. However, these methods do not consider the contextual/relational information of things and stuff when generating features and making predictions.

**Relational Contexts in Segmentation:** More recently, researchers have started to look into learning and exploiting relational information for segmentation. [12, 20, 51, 55] utilize an attention mechanism to capture pixel interdependencies over an image. Specifically, in [51], the authors propose Object-Contextual Representations (OCR), which captures the correlations between pixel-wise features and object region representations and has enabled state-of-the-art performance in semantic segmentation. However, OCR only considers entire class regions and cannot be used for object instances. In our work, we go beyond OCR and propose a novel method, PISR, to learn and capture relations among both object instances (things) and stuff. Our proposed method can be used with any existing panoptic segmentation architecture and can considerably enhance the segmentation performance, as we shall see in the paper. In [12], the authors build implicit relations based on transformer models using queries. Different from this, PISR is a general module to capture explicit relations among panoptic classes and the relations between each class and the scene. Furthermore, it reweights each component to focus on the more useful information and enhance any existing model, including transformer-based ones.

## 3. Proposed Approach

This section describes our novel Panoptic, Instance, and Semantic Relations (PISR) model that captures the relations from semantic and instance features to generate a final representation that is more informative for panoptic segmentation. We provide an overview of PISR and how it works with any given base architecture (e.g., an existing panoptic segmentation model) in Section 3.1, and then discuss its

<sup>1</sup>See [2, 9, 14, 17, 23, 31, 37, 39, 40, 43, 44] and [3, 4, 6–8, 16, 20, 33, 42, 51–55] for recent methods on instance and semantic segmentation respectively, [5] for using segmentation for other tasks, and [34] for an extensive survey.

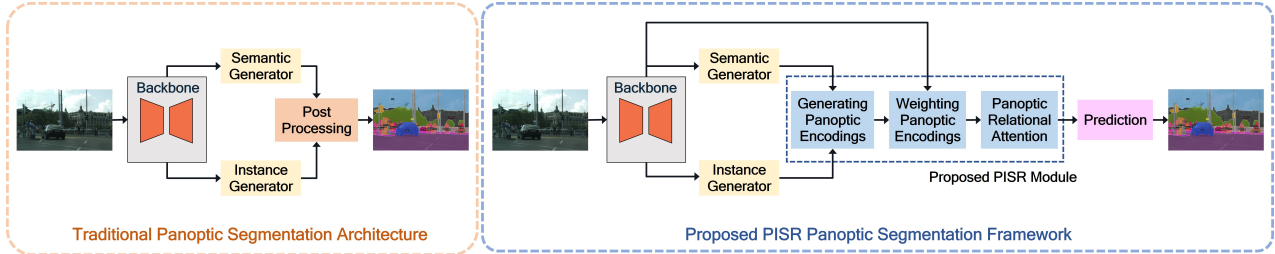


Figure 2. Left: Conventional panoptic segmentation architecture. Right: Our proposed Panoptic, Instance, and Semantic Relations (PISR) framework that can work with any base panoptic segmentation model.

components in detail, including how PISR generates the initial panoptic encodings, reweights them, and applies attention to capture key correlations, in Sections 3.2, 3.3, and 3.4.

### 3.1. Applying PISR for Panoptic Segmentation

A general panoptic segmentation architecture usually consists of four parts: 1) a backbone for feature extraction, 2) a semantic generator that outputs semantic segmentation, 3) an instance generator that outputs instance segmentation, and 4) a post-processing block that combines the two types of segmentations to produce the final panoptic segmentation. An illustration of this is given in Figure 2 (left). We refer to such a panoptic segmentation pipeline as a base architecture, which encompasses most existing methods.

PISR is a universal module and can be appended to any panoptic segmentation architecture. This not only includes traditional architectures [11, 21, 27] but also recent state-of-the-art models [12, 41]. First, it takes the outputs from the semantic (stuff) and instance (thing) generators. Only the top  $K$  most confident predictions are used to confine the instances to more reliable ones. We refer to each semantic class and each selected instance in these outputs as a panoptic category (e.g., the car class, person 1, person 2). PISR generates an initial encoding for each panoptic category, which summarizes the key features of the pixels assigned to that category. It then automatically reweights these initial encodings to highlight the more important ones. Next, the weighted panoptic encodings are fed into a Panoptic Relational Attention module, after which the enhanced features are sent to the prediction stage to generate the final panoptic segmentation. These steps are illustrated in Figure 2 (right).

When we do end-to-end training of an architecture with PISR, we apply the usual semantic and instance segmentation losses to the final estimated outputs. In addition, we also exert semantic and instance losses to the intermediate outputs from the semantic and instance generators that come with the base architecture. More formally, our total training loss function can be written as follows:

$$\mathcal{L} = \gamma \cdot (\mathcal{L}'_{sem} + \mathcal{L}'_{ins}) + \mathcal{L}_{sem} + \mathcal{L}_{ins}, \quad (1)$$

where  $\mathcal{L}_{sem}$  and  $\mathcal{L}_{ins}$  are the loss functions for predicting final instances and semantic segmentation respectively,  $\mathcal{L}'_{sem}$  and  $\mathcal{L}'_{ins}$  are the intermediate semantic and instance

loss functions and  $\gamma$  is treated as a hyperparameter. For both intermediate and final supervisions, we apply the same loss functions used to train each base architecture as reported in their original paper. For instance, when using Panoptic-DeepLab [11] as the base model, the semantic segmentation is supervised by a cross-entropy loss while the instance masks are supervised by an MSE loss for center heatmaps and an L1 loss for offsets. Training details and hyperparameters for each experiment are discussed in the Appendix.

### 3.2. Generating Initial Panoptic Encodings

Given the features supplied by the backbone, we generate panoptic encodings which summarize the key features of the semantic classes and selected instances. The procedures for generating panoptic encodings are shown in Figure 3 (a) and described in detail in the following.

**Semantic Encodings:** Suppose that the backbone network generates a feature map  $F \in \mathbb{R}^{C \times HW}$ , where  $C$ ,  $H$ , and  $W$  are the number of channels, height, and width of the feature map. Taking  $F$  as input, the semantic generator produces a soft semantic segmentation map  $S \in \mathbb{R}^{N_{sem} \times HW}$ , where  $N_{sem}$  is the number of semantic classes, and for each pixel location, a probability vector that indicates how likely this pixel belongs to different classes. We calculate the semantic encodings  $E_{sem} \in \mathbb{R}^{N_{sem} \times C}$  by multiplying  $S$  and  $F$ :  $E_{sem} = SF^T$ . These encodings contain the most prominent features for the semantic classes.

**Instance Encodings:** Standard instance predictions contain a center mass  $M \in \mathbb{R}^{1 \times HW}$  and a center offset  $O \in \mathbb{R}^{2 \times HW}$ .  $M$  is the objectness score, which we use to select the top- $K$  most confident center locations as shown in Figure 5 (e), (f). Given these  $K$  selected centers, we then produce  $K$  initial heatmaps based on their center offsets.<sup>2</sup> Next, we convert the predicted semantic segmentation  $S$  into a binary segmentation of things and stuff, and then multiply it with the initial heatmaps in order to suppress the background. The resulting instance heatmaps are denoted

<sup>2</sup>We generate the initial instance heatmap,  $H_{init} \in \mathbb{R}^{K \times HW}$ , by performing simple instance center regression:  $H_{inst}(i, j) = 1 - C(M(i, j) - (i + O_x(i, j), j + O_y(i, j)))$ , where  $C$  is a normalization constant and  $(i, j)$  is the pixel location. We provide further visualizations for these initial instance heatmaps in the Appendix.

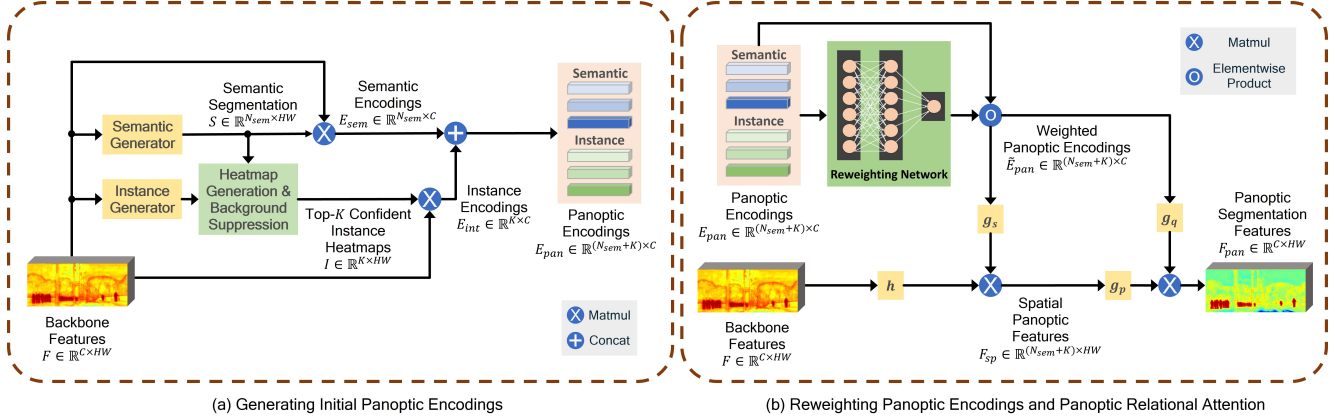


Figure 3. (a) Details on how the initial panoptic encodings are generated. (b) Details on how PISR reweights the initial encodings and subsequently applies two-stage attention by employing PRA to generate the final panoptic segmentation features.

as  $I \in \mathbb{R}^{K \times HW}$ . Finally, we calculate the instance encodings  $E_{ins} \in \mathbb{R}^{K \times C}$  by multiplying  $I$  and  $F$ :  $E_{ins} = IF^T$ .

**Panoptic Encodings:** The semantic encodings and instance encodings are concatenated to form the final panoptic encodings:  $E_{pan} \in \mathbb{R}^{(N_{sem}+K) \times C}$ . Each panoptic encoding then summarizes the key features of a semantic class or one of the selected instances.

### 3.3. Reweighting Panoptic Encodings

Given the panoptic encodings  $E_{pan}$ , we further reweigh them based on their importance. More specifically, we use a small 2-layer fully-connected network with a sigmoid output layer to generate the weights.<sup>3</sup> It takes  $E_{pan}$  as input and outputs the weight vector  $\omega \in \mathbb{R}^{(N_{sem}+K) \times 1}$ . Each element in  $\omega$  is a predicted importance score for a panoptic category. We then compute the weighted panoptic encodings as follows:  $\tilde{E}_{pan} = E_{pan} \circ [\omega]$ , where  $[\omega] \in \mathbb{R}^{(N_{sem}+K) \times C}$  is the broadcasted version of  $\omega \in \mathbb{R}^{(N_{sem}+K) \times 1}$  across the  $C$  dimension of  $E_{pan}$  while  $\circ$  is the element-wise product.

By doing this, PISR learns to focus on the important semantic classes and instances, while suppressing the less relevant ones. Although this reweighting network employs a simple structure, it provides a critical capability for PISR to be more robust to  $K$ . As we shall see in the experiments, as  $K$  increases, this reweighting allows PISR to generate improving panoptic segmentation whereas the performance can degrade without weighting. By analyzing output weights, we found that the reweighting network learns to down-weight classes that are not present in the scene and false-positive instances that make it into top- $K$  predictions.

### 3.4. Panoptic Relational Attention

Panoptic segmentation requires a holistic understanding of the scene, including both things and stuff. However, existing approaches do not fully utilize the relations among semantic classes and instances. To enable the network

<sup>3</sup>Larger reweighting networks provided no significant improvements.

to leverage the underlying relational contexts, we devise a Panoptic Relational Attention (PRA) module computing correlations across the panoptic categories based on panoptic encodings. PRA takes global features  $F$  and panoptic encodings  $\tilde{E}_{pan}$  as input. Two stages of attention are then applied to extract various types of correlations. The details of PRA are shown in Figure 3 (b) and discussed as follows.

First, we correlate the weighted panoptic encodings with the spatial features. This produces a spatial panoptic feature map:  $F_{sp} = g_s(\tilde{E}_{pan})^T h(F)$ , where  $F_{sp} \in \mathbb{R}^{(N_{sem}+K) \times HW}$ , and  $g_s$  and  $h$  contain  $1 \times 1$  and  $3 \times 3$  convolutional layers, respectively. This captures the panoptic signals in each pixel location. Next, we take the spatial panoptic feature map  $F_{sp}$  and correlate it with the weighted panoptic encodings  $\tilde{E}_{pan}$ . This produces the final panoptic segmentation features:  $F_{pan} = g_q(\tilde{E}_{pan})^T g_p(F_{sp})$ , where  $F_{pan} \in \mathbb{R}^{C \times HW}$  and  $g_p$  and  $g_q$  contain  $1 \times 1$  convolutional layers. This final feature map  $F_{pan}$  carries the enhanced panoptic signals over the spatial pixel locations and is fed to the final prediction stage to generate the semantic and instance segmentation, and the final panoptic segmentation.

## 4. Experiments

In this section (and also in the Supplementary File), we present comprehensive performance evaluations of PISR on large benchmark datasets, compare it with baselines and the current state of the art (SOTA), and conduct extensive ablation studies on various aspects of PISR.

### 4.1. Experimental Setup

**Datasets:** We evaluate performance on Cityscapes [13], COCO [29], and ADE20K [56] panoptic segmentation datasets. Cityscapes consists of 3,475 annotated images of size  $1024 \times 2048$ , with 2,975 training and 500 validation images. It covers 11 stuff and 8 thing classes. COCO consists of 118,000 training, 5,000 validation, and 20,000 test images, covering 53 stuff and 80 thing classes. ADE20K

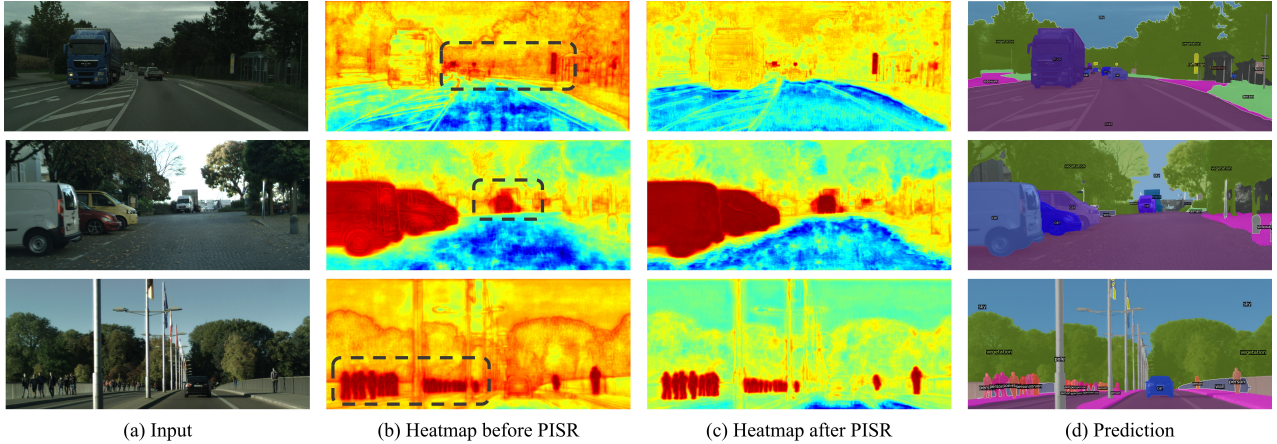


Figure 4. Visualization of the panoptic segmentation features before and after PISR, for three sample classes (top: traffic sign, middle: car, bottom: person). In the four columns we show: (a) input RGB image, (b) heatmap visualization for a given class before PISR, (c) heatmap visualization for the same given class after PISR, and (d) final panoptic segmentation. Darker red (lighter blue) colors indicate stronger (weaker) signals in the feature map. With PISR, shapes of the target-class objects are more accurately captured and their signals are considerably stronger. In regions where multiple instances interact with each other, PISR also captures clearer features as indicated by the sharp boundaries of instances such as people and cars. Sample improvements are highlighted by the boxes.

contains 20,210 training, 2,000 validation and 3,000 test images, with 35 stuff and 115 thing classes.

**Networks and Training:** We implement PISR with several state-of-the-art base panoptic segmentation architectures including Panoptic-DeepLab [11], Panoptic-FPN [21], and Maskformer [12]. We experiment with various backbones for these models, including ResNet-50, ResNet-101 [18], HRNet-w48 [42], Swin-L [32] and ResNet-50-FPN [21]. When training each base architecture with PISR, we use the original semantic and instance loss functions for both intermediate and final supervisions. We follow the original training settings, e.g., hyperparameters, training schedule, etc. Training details and hyperparameters are included in the supplementary material.

**Baseline:** In addition to comparing with the existing panoptic segmentation models, we also implement a strong baseline where the Object-Contextual Relationship (OCR) module [51] is applied to Panoptic-DeepLab. By comparing with this baseline, we directly show the advantage of leveraging both things and stuff relations in PISR, as compared to OCR which only considers semantic classes.

**Evaluation Metrics:** We use standard metrics to evaluate panoptic segmentation performance, including panoptic quality (PQ) [22], semantic quality (SQ), and recognition quality (RQ). We further report PQ scores for things and stuff, denoted as  $PQ^{th}$  and  $PQ^{st}$ , respectively. For semantic segmentation and instance segmentation predictions, we report mean Intersection over Union (mIoU) and Mask Average Precision (AP) [29] scores, respectively.

## 4.2. Heatmap Visualization

We show the visualization of the enhanced heatmaps by using our proposed PISR (column (c)), as well as the original ones without PISR (column (b)) in Figure 4. For each row, we visualize the heatmaps for a sample class. The heatmaps of all the selected instances of the same class are overlaid on the same image for better visualization.

It can be seen that PISR significantly improves the recognition of the target classes, as shown by the stronger signals depicted by darker red colors. We also observe that PISR recognizes sharper boundaries around regions where lots of instances are interacting with each other.

In Figure 5, we show visualizations for two Panoptic-DeepLab-ResNet50 models, trained with OCR and PISR, respectively. We overlay the obtained Semantic and Center (S+C) heatmaps for the “trains” category and present them together. We also show the final predictions obtained from both models. It is visible that OCR and PISR both generate similar semantic heatmaps; however, OCR predicts multiple centers for the “train” category compared to PISR, leading to poor instance segmentation results. In this case, even though both intermediate predictions are similar, PISR is able to effectively rectify the errors by incorporating instance information into its relational attention scheme.

## 4.3. Evaluation Results

**Cityscapes:** We report results on the Cityscapes-val split in Table 1 and Figure 6. We use Panoptic-DeepLab with various backbones as our base model, and train these networks using PISR. In the first part of Table 1, we compare with recent methods which use ResNet-50 or its variants as the backbone. It can be seen that our proposed PISR module significantly improves panoptic segmentation performance of the base model and also outperforms other pipelines. PISR enables much higher gains over the original Panoptic-DeepLab as compared to the baseline where

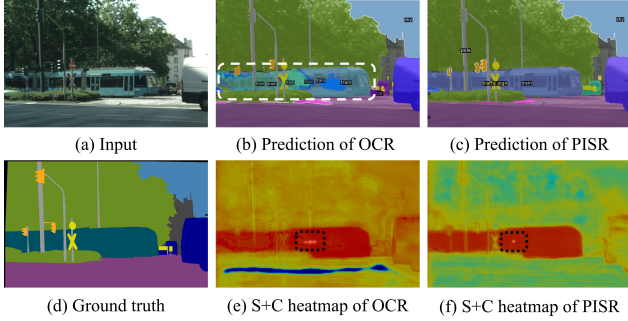


Figure 5. Visualization of the comparison between OCR and PISR. For (e) and (f), we visualize semantic heatmaps and instance center results overlaid together for the “train” class. Note that OCR predicts multiple overlapped centers, even when the number of instances of “train” class is only one. OCR does not consider instance features, hence fails to correct wrong prediction of instances. PISR on the other hand encompasses every semantic and instance feature by analyzing similarity among them, and thus successfully produces better instance masks and centers.

OCR is used. This is due to the fact that PISR considers the relations among both semantic classes and key instances, while OCR only utilizes semantic relations.

In the second section of Table 1, we compare with SOTA methods that use larger backbones (e.g., ResNet-101, EfficientNet). In this case, we use the HRNet-w48 backbone. Our PISR-enhanced Panoptic-DeepLab outperforms the SOTA methods. For fair comparison, we consider versions of these networks initialized using ImageNet pre-trained weights and we perform single-scale inference.<sup>4</sup>

Figure 6 shows qualitative results obtained by using Panoptic-DeepLab (ResNet-50) without and with PISR. It can be clearly seen from the highlighted regions that using PISR leads to overall more accurate segmentation as compared to the baseline prediction. For instance, PISR successfully exploits panoptic relations to predict the correct semantic label for “building” (and not “sky”) even though it contains a reflection of the sky.

**COCO:** Our results on COCO are summarized in Table 2. We apply PISR to different base architectures with various backbones and compare with recent SOTA methods. It can be seen that by adding PISR, we significantly enhance the panoptic segmentation performance of the base model. For instance, for Panoptic-DeepLab with ResNet-50 backbone, PISR increases the PQ score considerably from 35.5 to 38.8, and outperforming the option of using OCR, which has a PQ score of 37.2.

We also compare with the recent state-of-the-art works on both COCO val and test-dev splits. We trained a baseline UPerNet model modified for the panoptic segmentation

<sup>4</sup>Multi-scale inference is a computationally expensive technique that can in principle be applied to any network, including a PISR-enhanced model. In our evaluation, we do not apply multi-scale inference to ensure a fair comparison with all recent works which do not use it.

Method	Backbone	PQ	AP	mIoU
Panoptic FPN [21]	RN50-FPN	58.1	33.0	75.7
UPSNNet [48]	RN50-FPN	59.3	<b>33.3</b>	75.2
Unifying [25]	RN50-FPN	61.4	<b>33.3</b>	79.5
LPSNet [19]	RN50-FPN	60.4	33.0	78.6
Seamseg [36]	RN50-FPN	60.2	<b>33.3</b>	74.9
COPS [1]	RN50	62.1	-	-
Panoptic FCN [27]	RN50-FPN	61.4	-	-
AdaptIS [39]	RN50	59.0	32.3	75.3
Panoptic-DL [11]	RN50	59.9	32.1	78.5
Panoptic-DL [11] + OCR	RN50	60.7	32.1	79.6
Panoptic-DL [11] + PISR	RN50	<b>62.2</b>	<b>33.3</b>	<b>80.2</b>
AdaptIS [39]	RNX101	62.0	36.3	79.2
UPSNNet [48]	RN101	61.8	<b>39.0</b>	79.2
Panoptic FPN [21]	RN101-FPN	61.2	36.7	80.4
EfficientPS [35]	EffB5	63.9	38.3	79.3
Panoptic-DL [11]	RN101	60.5	33.7	79.0
Panoptic-DL [11]	X71	63.0	35.3	80.5
Panoptic-DL [11]	HR48	63.4	36.2	80.6
Panoptic-DL [11] + PISR	HR48	<b>64.1</b>	<b>37.6</b>	<b>80.7</b>

Table 1. Quantitative evaluation on the Cityscapes val set in terms of PQ, AP, and mIoU. We compare with existing methods using RN-50 (upper part) or other variants (lower part) as backbone. We also apply PISR to Panoptic-DeepLab with HRNet-w48 backbone and compare it with other existing methods which use larger backbones. We report the model performance for which the training is initialized with ImageNet pre-trained weights. RN, RNX, Eff, and HR48 refer to ResNet, ResNeXt, EfficientNet, and HRNet-w48, respectively. Gray rows are new models (ours) introduced in this paper. The best numbers in each section are highlighted in bold.

Split	Method	Backbone	PQ	PQ <sup>th</sup>	PQ <sup>st</sup>
Val	UPSNNet [48]	RN50-FPN	42.5	48.5	33.4
	AUNet [26]	RN50-FPN	39.6	<b>49.1</b>	25.2
	CIAE (640) [15]	RN50-FPN	39.5	44.4	33.1
	COPS [1]	RN50	38.4	40.5	35.2
	OANet [30]	RN50-FPN	39.0	48.3	24.9
	AdaptIS [39]	RN50	35.9	40.3	29.3
	SSAP [14]	RN50	36.5	40.1	32.0
	LPSNet [19]	RN50	39.1	43.9	30.1
	Panoptic-FPN [21]	RN50-FPN	39.2	46.6	27.9
	Panoptic-FPN + PISR	RN50-FPN	<b>42.7</b>	<b>48.7</b>	<b>33.6</b>
	Panoptic-DL [11]	RN50	35.5	37.8	32.0
	Panoptic-DL [11] + OCR	RN50	37.2	38.9	35.7
	Panoptic-DL [11] + PISR	RN50	<b>38.8</b>	<b>40.6</b>	<b>36.2</b>
	Panoptic-DL [11]	HR48	37.8	-	-
	Panoptic-DL [11] + PISR	HR48	<b>40.7</b>	<b>42.6</b>	<b>37.7</b>
	Panoptic-FCN [27]	Swin-L	52.1	58.5	42.3
	Pan-SegFormer [28]	PvTv2-B5	<b>54.1</b>	<b>60.4</b>	<b>44.6</b>
	Pan-SegFormer [28]	PvTv2-B2	52.6	58.2	43.3
	Max-DeepLab [41]	Max-L	51.1	57.0	42.2
	MaskFormer [12]	Swin-L	52.7	58.5	44.0
UPerNet [47]	Swin-L	50.3	55.7	42.1	
UPerNet [47] + PISR	Swin-L	<b>52.9</b>	<b>58.9</b>	<b>43.8</b>	
Test	Panoptic-FCN [27]	Swin-L	52.7	59.4	42.5
	Refine [38]	RNX101-FPN	51.5	59.6	39.2
	Max-DeepLab [41]	Max-L	51.3	57.2	43.4
	UPerNet [47]	Swin-L	50.9	56.7	42.3
	UPerNet [47] + PISR	Swin-L	<b>53.2</b>	<b>59.2</b>	<b>44.2</b>

Table 2. Quantitative evaluation on the COCO validation and test sets, in terms of PQ, PQ<sup>th</sup>, and PQ<sup>st</sup>. RN and HR48 indicate ResNet and HRNet-w48, respectively. Gray rows are our models introduced in this paper. Best numbers are highlighted in bold.

task. We then compare with an UPerNet model trained after adding the PISR block. We obtain a PQ score of 52.7, outperforming the baseline by 2.4 PQ. As seen from the numbers, our approach obtains comparable performance with recent SOTA methods on both val and test splits, and ranks

Method	Backbone	PQ	PQ <sup>Th</sup>	PQ <sup>St</sup>	SQ	RQ
Panoptic-FCN [27]	RN50	30.1	34.1	27.3	-	-
BGRNet [45]	RN50	31.8	34.1	27.3	-	-
Auto-pan. [46]	SV2	32.4	33.5	30.2	-	-
MaskFormer [12]	RN50	34.7	32.5	38.0	76.3	41.7
MaskFormer + PISR	RN50	36.1	34.7	39.0	78.3	44.3
MaskFormer	RN101	35.7	34.5	38.0	77.4	43.8
MaskFormer + PISR	RN101	<b>37.0</b>	<b>35.6</b>	<b>39.7</b>	<b>79.9</b>	<b>45.2</b>

Table 3. Quantitative evaluation on the ADE20K validation set. RN and SV2 indicates ResNet and ShuffleNetV2, respectively. Gray rows are new models (ours) introduced in this paper. Best numbers are highlighted in bold.

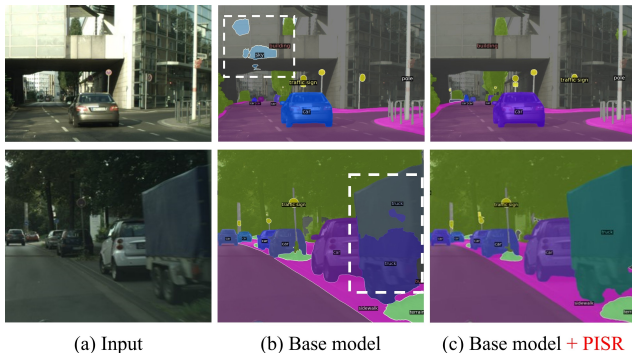


Figure 6. Qualitative results on Cityscapes: (a) Input images. (b) Predictions by using Panoptic-DeepLab (ResNet-50). (c) Our results by applying PISR to Panoptic-DeepLab (ResNet-50). The overall panoptic segmentation quality improves with PISR. Dashed boxes highlight sample regions where PISR significantly enhances the baseline prediction.

amongst the top methods on the public leaderboard.

**ADE20K:** We further evaluate PISR on ADE20K. It can be seen in Table 3 that by applying PISR to MaskFormer, which has SOTA performance on ADE20K, we improved its accuracy further. For instance, when using the ResNet-101 backbone, PISR significantly improves the PQ score from 35.7 to 37.0.

Figure 7 shows qualitative results obtained by using MaskFormer (ResNet-50) without and with PISR. Overall, PISR enhances the segmentation quality with clearer boundaries and more complete object masks (e.g., the towel in the bottom-row example). In addition, PISR has the capability to rectify masks that are entirely wrong. For instance, the tray in the top-row example is misclassified as a portrait when using the baseline alone (highlighted by a box), due to the image of people on it. By analyzing the instance relations, PISR successfully predicts that it is actually a tray. It is also visible that PISR can accurately segment the drawer, which was misclassified by the base model.

#### 4.4. Accuracy-Computation Analysis

In Table 4, we report the effect on computations (in GFLOPS) v/s accuracy (in PQ) by adding PISR for MaskFormer and Panoptic-DeepLab respectively. We see that PISR increases baseline computations but provides a much

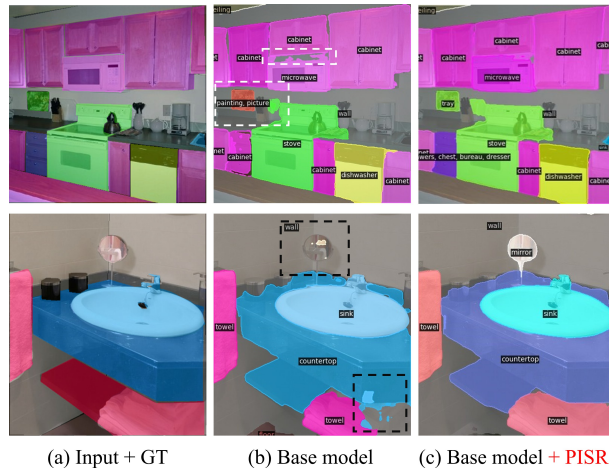


Figure 7. Qualitative results on ADE20K: (a) Input images with ground-truth masks overlaid. (b) Predictions by using MaskFormer (ResNet-50). (c) Our results by applying PISR to MaskFormer (ResNet-50). When using PISR, the overall panoptic segmentation quality improves. Dashed boxes highlight sample regions where PISR significantly enhances the baseline prediction.

better accuracy gain as compared to simply scaling up the backbone. For instance, when switching from ResNet-50 to ResNet-101, Maskformer increases its PQ from 34.7 to 35.7 on ADE20K, with GFLOPS increased from 116.6 to 159.3 and throughput decreased from 21.1 FPS to 19.6 FPS. On the other hand, by adding PISR to Maskformer with ResNet-50, we achieve a higher PQ of 36.1, incur a smaller GFLOPS count of 136.0 and only slightly lower throughput of 20.4 FPS. Similarly, for Panoptic-DeepLab, using PISR allows more accuracy gain and has less computation increase as compared to scaling up the backbone.

Figure 8 provides a graphical illustration on using different ways to enhance panoptic segmentation accuracy by adding computation. The grey curve shows the effect of simply scaling up the baseline network. The blue dot shows the accuracy and computation resulted from applying the OCR module. The red curve shows the accuracy and computation of using PISR, by varying the intermediate number of channels in the PISR block. It can be seen that PISR provides a much better accuracy-computation trade-off as compared to scaling up the backbone or using OCR. We also show the effect of using PISR but without reweighting the encodings. While this has a lower accuracy and uses less computation, it still provides a favorable trade-off.

#### 4.5. Ablation Studies

**Comparing with Other Feature Processing Schemes:** We implement and compare with other ways of processing semantic and instance features to generate the final features for panoptic segmentation. In this study, we use Panoptic-DeepLab (ResNet-50) as the base model and report results on the Cityscapes dataset. We consider the following al-

Dataset	Method	Backbone	PQ	GFLOP	FPS
ADE20K (375 X 500)	MaskFormer [12]	RN50	34.7	116.6	21.1
	MaskFormer + PISR	RN50	36.1	136.0	20.4
	MaskFormer	RN101	35.7	159.3	19.6
Cityscapes (1024 X 2048)	Panoptic-DL [11]	RN50	59.7	419.5	7.6
	Panoptic-DL + PISR	RN50	62.2	575.0	6.8
	Panoptic-DL	RN101	60.5	575.5	6.5

Table 4. Accuracy (in PQ), computational complexity (in GFLOPS) and throughput (in FPS measured on Tesla-A100 GPU) for Maskformer and Panoptic-DeepLab, on ADE20K val and Cityscapes val, respectively. It can be seen that using PISR provides more accuracy improvement and increases less computation as compared to simply using a heavier backbone.

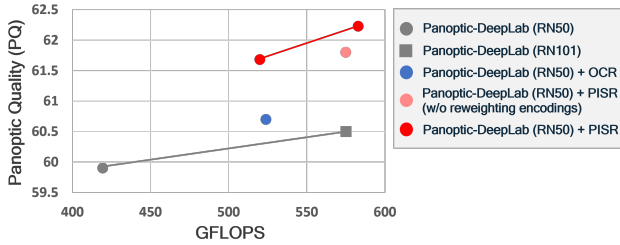


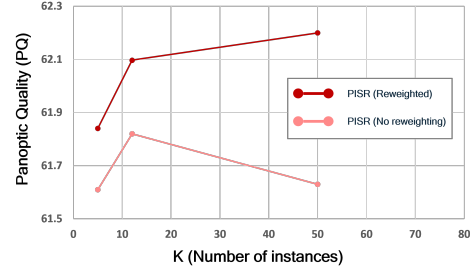
Figure 8. Panoptic Quality vs. GFLOPS on Cityscapes, using Panoptic-DeepLab as the base model. We show results for the base model with different backbones, base model + OCR, base model + PISR (different model sizes), and base model + PISR without reweighting encodings. It can be seen that using PISR enables a much better accuracy-computation trade-off.

Model	PQ	mIoU	iIoU	$\Delta$
Panoptic-DL [11]	59.9	78.5	62.5	0
Panoptic-DL + Concatenation	60.6	79.1	62.6	+1.4
Panoptic-DL + Elementwise Product	60.4	78.2	63.0	+0.7
Panoptic-DL + OCR	60.7	79.5	62.4	+1.7
Panoptic-DL + PISR (w/o Reweighting)	61.8	79.6	64.0	+4.5
Panoptic-DL + PISR	<b>62.2</b>	<b>80.2</b>	<b>64.4</b>	<b>+5.9</b>

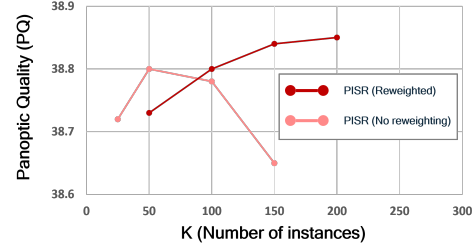
Table 5. Comparing alternative ways to process semantic and instance features.  $\Delta$  is the sum of all gains in PQ, mIoU, and iIoU.

ternative options: 1) *Concatenation*: We simply concatenate  $S$ ,  $I$ , and  $F$ , and use the stacked features for the final prediction. 2) *Elementwise Product*:  $S$  and  $I$  are first concatenated and passed through a convolutional layer to match the dimensions of  $F$ . The elementwise product of the resulting tensor and  $F$  is used as the final features. 3) *OCR [51]*: We apply the OCR module to the semantic features. The instance features cannot be used as OCR cannot handle a variable number of predicted instances. Table 5 summarizes the results. It can be seen that concatenation, elementwise product, and OCR brings minor improvements over the baseline as they do not properly capture the semantic and instance relations. In contrast, PISR enables a larger performance gain by properly encoding the relations.

**Effect of  $K$** : We study the effect of varying  $K$  on the final PQ for both Cityscapes and COCO in Figure 9. The pink curves show the performance of PISR with Panoptic-DeepLab (ResNet-50). It can be seen that the PQ scores improve up to an optimum as  $K$  increases, then the gain



(a) Top- $K$  analysis on Cityscapes val



(b) Top- $K$  analysis on COCO val

Figure 9. Observing PQ w.r.t. various values of  $K$  on Cityscapes and COCO.

flattens out when  $K$  becomes larger.

**Effect of Reweighting**: As shown in Table 5, applying our reweighting module provides additional gains compared to the case without reweighting. Although we use a small reweighting network, it provides a critical capability for PISR to focus on the important encodings rather than treating them all the same way. We apply the reweighting module at various  $K$  and study the effect on PQ in Figure 9. As observed from the red curves, our reweighting module makes PISR more robust to  $K$ . PISR better learns to utilize only the useful information and does not suffer a heavy performance drop as  $K$  increases (red curves in Figure 9). This also widens out the search-space greatly for  $K$  as a hyper-parameter. Without reweighting, the segmentation performance drops when  $K$  becomes large since unreliable information might be permeated (pink curves in Figure 9).

## 5. Conclusion

We propose a novel method called PISR to combine both semantic and instance information for panoptic segmentation. Our method consists of learning a general module that can be applied on top of existing panoptic segmentation architectures. The module captures interactions among instance and stuff embeddings to produce a context-aware segmentation map. Furthermore, a weighing scheme is introduced to produce varying contributions of different instances in the scene. Experimental evaluation on three large standard datasets for panoptic segmentation show that PISR improves segmentation performance over existing methods. We also conduct additional ablation studies and analyses to better understand our framework.



## References

- [1] A. Abbas and P. Swoboda. Combinatorial optimization for panoptic segmentation: A fully differentiable approach. *arXiv preprint arXiv:2106.03188*, 2021. 6
- [2] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019. 2
- [3] S. Borse, H. Cai, Y. Zhang, and F. Porikli. Hs3: Learning with proper task complexity in hierarchically supervised semantic segmentation. 2021. 2
- [4] S. Borse, Y. Wang, Y. Zhang, and F. Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5911, 2021. 2
- [5] H. Cai, J. Matai, S. Borse, Y. Zhang, A. Ansari, and F. Porikli. X-distill: Improving self-supervised monocular depth via cross-task distillation. In *Proceedings of the British Machine Vision Conference*, 2022. 2
- [6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *International Conference on Learning Representations*, 2015. 2
- [7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2
- [8] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. arxiv 2017. *arXiv preprint arXiv:1706.05587*, 2019. 2
- [9] X. Chen, R. Girshick, K. He, and P. Dollár. Tensormask: A foundation for dense object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2061–2069, 2019. 2
- [10] Y. Chen, G. Lin, S. Li, O. Bourahla, Y. Wu, F. Wang, J. Feng, M. Xu, and X. Li. BANet: bidirectional aggregation network with occlusion handling for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2020. 2
- [11] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020. 2, 3, 5, 6, 8
- [12] B. Cheng, A. G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems*, 2021. 2, 3, 5, 6, 7, 8
- [13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 4
- [14] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 642–651, 2019. 2, 6
- [15] N. Gao, Y. Shan, X. Zhao, and K. Huang. Learning category-and instance-aware pixel embedding for fast panoptic segmentation. *IEEE Transactions on Image Processing*, 30:6013–6023, 2021.
- [16] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European conference on computer vision*, pages 519–534. Springer, 2016. 2
- [17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [19] W. Hong, Q. Guo, W. Zhang, J. Chen, and W. Chu. Lpsnet: A lightweight solution for fast panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16746–16754, June 2021. 6
- [20] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 2
- [21] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 2, 3, 5, 6
- [22] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 1, 2, 5
- [23] Y. Lee and J. Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 2
- [24] Q. Li, A. Arnab, and P. H. Torr. Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 102–118, 2018. 2
- [25] Q. Li, X. Qi, and P. H. 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. 2, 6
- [26] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7026–7035, 2019. 2
- [27] Y. Li, H. Zhao, X. Qi, L. Wang, Z. Li, J. Sun, and J. Jia. Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2021. 2, 3, 6, 7

- [28] Z. Li, W. Wang, E. Xie, Z. Yu, A. Anandkumar, J. M. Alvarez, T. Lu, and P. Luo. Panoptic segformer. *arXiv preprint arXiv:2109.03814*, 2021. 6
- [29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4, 5
- [30] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6172–6181, 2019.
- [31] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 2
- [32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 5
- [33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2
- [34] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2
- [35] R. Mohan and A. Valada. Efficientpts: Efficient panoptic segmentation. *International Journal of Computer Vision (IJCV)*, 2021. 6
- [36] L. Porzi, S. Rota Bulò, A. Colovic, and P. Kotschieder. Seamless scene segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6
- [37] L. Qi, Y. Wang, Y. Chen, Y.-C. Chen, X. Zhang, J. Sun, and J. Jia. Pointnets: Point-based instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2
- [38] J. Ren, C. Yu, Z. Cai, M. Zhang, C. Chen, H. Zhao, S. Yi, and H. Li. Refine: Prediction fusion network for panoptic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2477–2485, 2021. 6
- [39] K. Sofiiuk, O. Barinova, and A. Konushin. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7355–7363, 2019. 2, 6
- [40] Z. Tian, C. Shen, and H. Chen. Conditional convolutions for instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 282–298. Springer, 2020. 2
- [41] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021. 3, 6
- [42] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2, 5
- [43] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, pages 649–665. Springer, 2020. 2
- [44] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen. Solov2: Dynamic and fast instance segmentation. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2
- [45] Y. Wu, G. Zhang, Y. Gao, X. Deng, K. Gong, X. Liang, and L. 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. 7
- [46] Y. Wu, G. Zhang, H. Xu, X. Liang, and L. Lin. Auto-panoptic: Cooperative multi-component architecture search for panoptic segmentation. *Advances in Neural Information Processing Systems*, 33, 2020. 7
- [47] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 6
- [48] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 2, 6
- [49] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen. Deeplab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019. 2
- [50] Y. Yang, H. Li, X. Li, Q. Zhao, J. Wu, and Z. Lin. Sognet: Scene overlap graph network for panoptic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12637–12644, 2020. 2
- [51] Y. Yuan, X. Chen, X. Chen, and J. Wang. Segmentation transformer: Object-contextual representations for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, volume 1, 2020. 2, 5, 8
- [52] Y. Zhang, S. Borse, H. Cai, and F. Porikli. Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2339–2348, 2022. 2
- [53] Y. Zhang, S. Borse, H. Cai, Y. Wang, N. Bi, X. Jiang, and F. Porikli. Perceptual consistency in video segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2564–2573, 2022. 2
- [54] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2
- [55] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018. 2

- [56] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 4