Point-supervised Panoptic Segmentation via Estimating Pseudo Labels from Learnable Distance

Jing Li[1,2,3,5], Junsong Fan[4] ✉, and Zhaoxiang Zhang[1,2,3,4,5] ✉

[1] University of Chinese Academy of Sciences (UCAS), Beijing, China
[2] New Laboratory of Pattern Recognition (NLPR), Beijing, China
[3] Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China
[4] Centre for Artificial Intelligence and Robotics, HKISI_CAS, HongKong, China
[5] State Key Laboratory of Multimodal Artificial Intelligence Systems, Beijing, China
{lijing2018, junsong.fan, zhaoxiang.zhang}@ia.ac.cn

Abstract. To bridge the gap between point labels and per-pixel labels, existing point-supervised panoptic segmentation methods usually estimate dense pseudo labels by assigning unlabeled pixels to corresponding instances according to rule-based pixel-to-instance distances. These distances cannot be optimized by point labels end to end and are usually suboptimal, which result in inaccurate pseudo labels. Here we propose to assign unlabeled pixels to corresponding instances based on a learnable distance. Specifically, we represent each instance as an anchor query, then predict the pixel-to-instance distance based on the cross-attention between anchor queries and pixel features through a distance branch, the predicted distance is supervised by point labels end to end. In order that each query can accurately represent the corresponding instance, we iteratively improve anchor queries through query aggregating and query enhancing processes, then improved distance results and pseudo labels are predicted with these queries. We have experimentally demonstrated the effectiveness of our approach and achieved state-of-the-art results.

Keywords: Weakly supervised learning · Panoptic segmentation · Point label

1 Introduction

Panoptic segmentation involves dividing an image into distinct masks for both thing and stuff categories, as described in [13]. Recently, deep learning-based models have shown great performance in classification [8,12], object detection [2, 30–32,45], and segmentation [15,16,34] tasks. Many deep learning-based panoptic segmentation methods [4,20,44] have been proposed, but their effectiveness relies on the availability of pixel-wise labels, and annotating these labels is a time-consuming process, which limits the widespread adoption of these methods in practical applications.

To address the challenge of heavy annotation, several approaches [10, 17–19, 35] suggest training panoptic segmentation models using dense pseudo labels

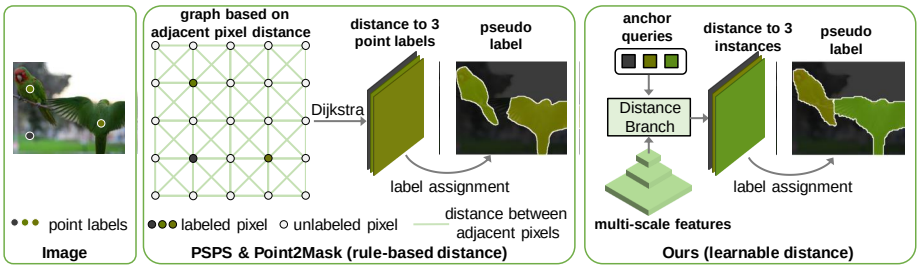

Fig. 1: PSPS [10] and Point2Mask [18] first compute distances between adjacent pixels based on specified features, then build a graph and get the distance between non-adjacent pixels through the Dijkstra algorithm, finally assign unlabeled pixels to appropriate instances based on the distance result, the distance can not be optimized by point labels. Our method predicts the pixel-to-instance distance directly through a distance branch, our distance results can be supervised by point labels end to end.

-7mm

estimated from image tags [35], bounding boxes [17], single- point labels [10,18], and multi-point labels [19]. Among these weak labels, the single-point label has attracted a large interest recently, because its annotation time is only marginally above the image tags [10], while spatial cues to distinguish different instances are provided by it. Consequently, this paper concentrates on achieving great panoptic segmentation performance with single-point labels (a single point label for each target).

Recently, PSPS [10] and Point2Mask [18] train the panoptic head with pseudo labels estimated from single-point labels and achieve satisfactory performance. PSPS [10] and Point2Mask [18] both estimate pseudo labels by assigning unlabeled pixels to corresponding instances according to rule-based pixel-to-instance distances (namely distances from pixel to point labels, each instance is represented by its point labels). As shown in Fig. 1, they first build a graph by taking pixels as vertices, and adjacent pixel distances as edges, then compute the distance between non-adjacent pixels through the Dijkstra algorithm. The distance between adjacent pixels is computed based on several features (semantic segmentation results, manifold features, Sobel edges, or contour results). The errors in these features may lead to wrong distance values, which cannot be revised by point label supervision and result in inaccurate pseudo labels.

To overcome the drawback of the rule-based distance, we propose an EPLD framework, which estimates pseudo labels from learnable distances. Specifically, we represent each instance as an anchor query and predict the pixel-to-instance distance based on the cross-attention between anchor queries and pixel features through a distance branch. Then, we assign each unlabeled pixel to its nearest instance and get the dense pseudo label. Since our distance result is learned rather than based on predefined rules (computing distances based on specified features and the Dijkstra algorithm), our distance is more accurate and produces better pseudo labels than previous methods.

Our distance branch is specifically designed for distance prediction in three aspects. Firstly, the number of instances in an image is variable, we initialize an anchor query for each instance adaptively by aggregating features at labeled pixels. Then we can predict a variable number of distance maps based on these anchor queries. Many DETR [3] based panoptic segmentation methods [4, 6, 20] feed a fixed number of queries to their decoder to predict segmentation results, these queries are not suitable for distance map estimation because they can not accurately correspond to a variable number of instances one by one like our anchor queries. Secondly, the initial anchor queries may be biased to instances' contour parts because point labels are sometimes at instance contour regions. We alleviate this problem by updating anchor queries iteratively through the query aggregating process. Thirdly, we further enhance anchor queries with instance class labels through a query enhancing process to help the distance branch estimate distance results utilizing the class cues. The main contributions of the paper are summarized as follows:

- We propose to estimate pseudo panoptic labels based on a learnable distance, which is more accurate than the rule-based distance in previous methods.
- We design a distance branch that is empowered by query aggregating and enhancing processes. It can estimate the pixel-to-instance distance accurately.
- We conduct experiments to demonstrate the effectiveness of our method. Our method achieves the new state-of-the-art performance (68.5% PQ on Pascal VOC and 41.0% PQ on COCO) with single-point labels as supervision.

2 Related Works

2.1 Panoptic Segmentation

Panoptic segmentation [13] involves the integration of semantic segmentation and instance segmentation to assign both a semantic class label and an instance ID label to each pixel in the input image. To address the challenges associated with this task, [13] proposes directly combining the results of semantic segmentation and instance segmentation. OANet [25] tackles occlusion issues between different instances by introducing a spatial ranking module. Knet [44] proposes to segment both thing and stuff regions with adaptively updated kernels. More recently, following transformer-based detection models DETR [3] and DeformableDETR [46], several transformer-based panoptic segmentation models have emerged. Panoptic SegFormer [20] utilizes two distinct query sets to represent thing and stuff contents, Mask2former [4] improves results by constraining the cross-attention within predicted mask regions to extract localized features.

2.2 Weakly Supervised Panoptic Segmentation

To alleviate the annotation burden of pixel-wise panoptic labels, some approaches have opted for weak labels, including image tags, bounding boxes, and points, as training labels. In [17], stuff regions are supervised by image tags, while instance

regions are supervised by bounding boxes. JTSM [35] generates pseudo masks for thing and stuff targets solely based on image tags. PSIS [5] supervises segmentation models with points sampled within box labels. PanopticFCN [19] employs a strategy of assigning multiple point labels to a single target and subsequently connecting these points to form polygon masks. PSPS [10] assigns label information from labeled points to unlabeled pixels based on a minimal traversing distance algorithm. Point2Mask [18] further improves PSPS by adopting a global optimal transportation strategy in the minimal traversing distance algorithm.

2.3 Point Labels for Segmentation

Recently point-based supervision has attracted more attention in the segmentation-related area [1,5,10,19,21]. Compared with scribbles [22,37–39,42] and boxes [40], point labels require much less time and effort to obtain and generate, while still including promising supervision to accomplish the model training, thus point labels have been widely explored in semantic segmentation task [1, 10, 21, 29]. Besides, point information is also a key foundation of the interactive segmentation task [24, 28, 43].

2.4 Cross Attention

Cross attention is a powerful mechanism used in various natural language processing (NLP) and computer vision applications. Originating from Transformer [41], cross attention operates between distinct sequences, enabling the model to align and integrate information from multiple sources. In NLP, this mechanism helps many models like BERT [7] and GPT [33] to handle complex dependencies between tokens in text sequences. In computer vision, this mechanism helps many models [3, 20, 46] to combine multi-scale or multi-source features and capture fine-grained contextual information. In this paper, we estimate pixel-to-instance distances based on the cross-attention maps between pixel features and instance anchor queries.

3 Approach

Since manually annotating per-pixel panoptic labels is labor-intensive, we propose to train panoptic segmentation models with point panoptic labels, which offer the instance class label $\mathcal{Y}$ and point-level instance ID mask $\mathcal{P}$ (we refer to one thing object or stuff class as one instance in this paper). To bridge the gap between point labels and per-pixel labels, our EPLD framework first estimates dense pseudo labels from point labels and then trains the panoptic models with pseudo labels. Next, we will elaborate on our EPLD framework and each component of EPLD.

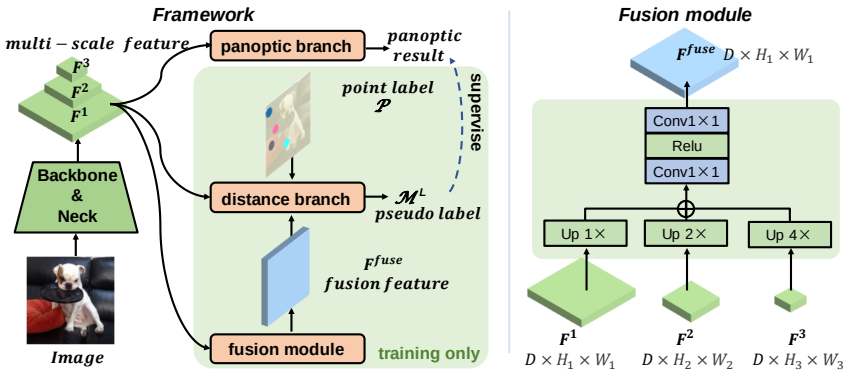

Fig. 2: Left: our EPLD framework. Right: our fusion module.

3.1 EPLD Framework

As shown in Fig. 2, during the training process, our EPLD framework first encodes the image as multi-scale features F^1, F^2, F^3 with the backbone and neck module, then feeds them to three modules: the panoptic branch, the distance branch, and the fusion module. The fusion module fuses F^1, F^2, and F^3 as fusion feature F^{fuse} which is fed to the distance branch. The distance branch estimates dense pseudo labels $\mathcal{M}^L$ from point labels $\mathcal{P}$ utilizing F^1, F^2, F^3, and F^{fuse}. The panoptic branch is supervised by the pseudo label $\mathcal{M}^L$.

During the inference process, the fusion module and the distance branch are discarded and don't incur any additional memory or computation cost, the model architecture and inference process are the same as the fully supervised model [20].

3.2 Fusion Module

The fusion module fuses multi-scale features F^1, F^2, and F^3 into one fusion feature map F^{fuse}. As shown in Fig. 2, for an input image with size $H \times W$, F^1, F^2, and F^3 are of size $D \times H_1 \times W_1$, $D \times H_2 \times W_2$, and $D \times H_3 \times W_3$, respectively, where $H_i = \frac{H}{2^{i+2}}$, $W_i = \frac{W}{2^{i+2}}$, D is the feature dimension. The fusion module first upsamples F^2 and F^3 to the size of $H_1 \times W_1$ and sums them with F^1, then feeds the summed feature to two convolution layers to produce F^{fuse} of size $D \times H_1 \times W_1$.

3.3 Distance Branch

As shown in Fig. 3, the distance branch contains L decoder layers and predicts the pseudo label iteratively with these layers. In each iteration, anchor queries are generated through the query aggregating process and further enhanced through the query enhancing process, then a decoder layer predicts the distance map based on the enhanced anchor queries to estimate the pseudo label. In the first

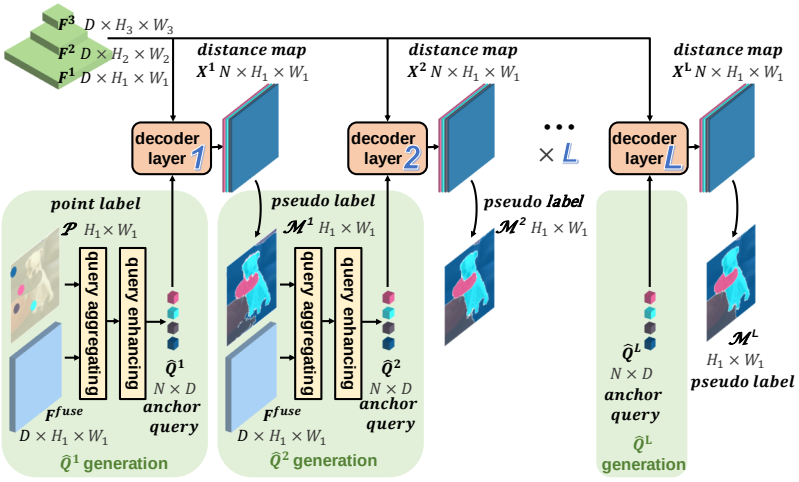

Fig. 3: Our distance branch contains L decoder layers. The distance map is predicted iteratively based on anchor queries. In each iteration, the anchor query is generated through query aggregating and query enhancing processes. The initial anchor query $\hat{Q}^1$ is aggregated using the point label $\mathcal{P}$, other anchor queries are aggregated using pseudo labels from the previous iteration. Please see Sec. 3 for more information.

iteration, the N anchor queries are generated by aggregating the fusion feature at labeled pixels of the point label $\mathcal{P}$. Since some point labels are at instance corner or edge regions, the aggregated queries may be biased to these local regions. Thus we aggregate new anchor queries in the second iteration according to pseudo labels estimated in the first iteration, these pseudo labels cover more complete instance regions and help to generate less biased anchor queries. Improved pseudo labels can be estimated with these less biased queries. To further improve the pseudo label, we aggregate new anchor queries with new pseudo labels for another $L-2$ iterations. Finally, we get the pseudo label $\mathcal{M}^L$ from the last iteration, we use $\mathcal{M}^L$ to train the panoptic branch. Next, we will elaborate on the query aggregating process, query enhancing process, distance map prediction, and decoder layer.

Query Aggregating Process. In the first iteration, $Q^1 \in \mathbb{R}^{N\times D}$ is aggregated using the point-level instance ID mask $\mathcal{P} \in \{1, 2, \cdots, N, 255\}^{H\times W}$, where N is the number of ground truth instances in the input image, pixels with 255 are unlabeled pixels, pixels with $n \in \{1, 2, \cdots, N\}$ belong to the n-th instance. We first transform $\mathcal{P}$ into the one-hot format $\mathcal{P}^{01} \in \{0, 1\}^{N\times H\times W}$, for unlabeled pixels, all N channels of $\mathcal{P}^{01}$ are set as zero. Then Q^1 is generated with $\mathcal{P}^{01}$ as follows:

$$Q_n^1 = \frac{1}{Z_n^1} \sum_{i=1}^{HW} \mathcal{P}_n^{01}[i] \cdot F^{fuse}[i] \tag{1}$$

where n denotes the n-th channel of $\mathcal{P}^{01}$ and n-th query item in Q^1, i is the pixel index, $Z_n^1 = \sum_{i=1}^{HW} \mathcal{P}_n^{01}[i]$ is the normalizing factor.

In other iterations, $Q^{l+1} \in \mathbb{R}^{N \times D}$ $(2 \leq l{+}1 \leq L)$ is aggregated using the pseudo label $\mathcal{M}^l \in \{1, 2, \cdots, N\}^{H \times W}$. We first transform $\mathcal{M}^l$ into the one-hot format $\mathcal{M}^{l\text{-}01}$ including N binarized maps, then apply erosion morphology operation to $\mathcal{M}^{l\text{-}01}$ to remove the noisy labels at contour regions and get $\widetilde{\mathcal{M}}^{l\text{-}01}$. Some instances depicted by point label $\mathcal{P}^{01}$ may be missing in $\widetilde{\mathcal{M}}^{l\text{-}01}$ after the erosion process, thus we revise these errors with $\mathcal{P}^{01}$ and get $\widehat{\mathcal{M}}^{l\text{-}01}$ as follows:

$$\widehat{\mathcal{M}}_n^{l\text{-}01}[i] = \begin{cases} 1 & \text{if } \mathcal{P}_n^{01}[i] = 1 \\ \widetilde{\mathcal{M}}_n^{l\text{-}01}[i] & \text{otherwise} \end{cases} \tag{2}$$

finally, we can get Q^{l+1} by aggregating fusion features as follows:

$$Q_n^{l+1} = \frac{1}{Z_n^{l+1}} \sum_{i=1}^{HW} \widehat{\mathcal{M}}_n^{l\text{-}01}[i] \cdot F^{fuse}[i] \tag{3}$$

where n denotes the n-th channel of $\widehat{\mathcal{M}}^{l\text{-}01}$ and n-th query item of Q^{l+1}, and i is the pixel index, Z_n^{l+1} is the normalizing factor.

Query Enhancing Process. After generating Q^l $(1 \leq l \leq L)$ through the aggregating process, we further enhance Q^l with class label $\mathcal{Y} \in \{1, 2, \cdots, C\}^N$ of N instances, where C is the total semantic class number of the dataset. We first apply an FC layer on Q^l to predict the class probability $Y^l \in [0, 1]^{N \times C}$, and supervise Y^l with $\mathcal{Y}$ to enhance Q^l implicitly:

$$\mathcal{L}_{cls}^l = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_{focal}(Y_n^l, \mathcal{Y}_n), \tag{4}$$

where $\mathcal{L}_{focal}$ is the focal loss, n is the instance index. Then we train C class embeddings $E^{cls} \in \mathbb{R}^{C \times D}$ representing C classes and enhance Q^l explicitly to get $\widehat{Q}^l$ as follows:

$$\widehat{Q}_n^l = Q_n^l + E^{cls}[\mathcal{Y}_n], \tag{5}$$

where n denotes the n-th instance, $E^{cls}[\mathcal{Y}_n]$ is the class embedding selected by the class label of the n-th instance. We feed $\widehat{Q}^l$ to the l-th decoder layer.

Distance Map Prediction. As shown in Fig. 3, the l-th decoder layer predicts a learnable distance map X^l of size $N \times H_1 \times W_1$, the n-th channel of X^l depicts the distance from $H_1 \times W_1$ pixels to the n-th instance. We supervise X^l with the point-level instance ID mask $\mathcal{P}$ by formulating the prediction of X^l as an instance classification problem for each pixel. Specifically, we let the l-th layer predict a N-channel probability map $M^l \in [0, 1]^{N \times H_1 \times W_1}$, which depicts the probability of assigning each pixel to N instances, then we supervise M^l with $\mathcal{P}$,

which offers the instance assigning label of several pixels. The detailed prediction and optimization process of M^l is in "**Decoder Layer**" in the following. Based on M^l, we can get X^l as follows:

$$X^l[i] = 1 - M^l[i], \tag{6}$$

where i is the pixel index. $M^l[i]$ depicts the probability of assigning the i-th pixel to N different instances. Intuitively, the greater the probability of assigning pixel i to an instance, the smaller its distance from this instance, thus $1 - M^l[i] \in [0,1]^N$ can be seen as a kind of distance from pixel i to N instances. With X^l, we generate the pseudo label $\mathcal{M}^l$ by assigning pixels to their nearest instances:

$$\mathcal{M}^l[i] = \arg\min_n X^l_n[i] = \arg\max_n M^l_n[i], \tag{7}$$

where n is the channel index, i is the pixel index.

Decoder Layer. Our distance branch contains L decoder layers with the same architecture, the l-th layer predicts the probability map M^l to estimate the distance map X^l and pseudo label $\mathcal{M}^l$ as described in "**Distance Map Prediction**" above. Following [20], we predict the probability map M^l based on the cross-attention maps between anchor queries $\widehat{Q}^l$ and multi-scale pixel features F^1, F^2, F^3, the detailed description of the decoder layer architecture can be seen in Sec. A.2 of the supplementary material.

To optimize M^l, we first supervise M^l with $\mathcal{P}$ through cross-entropy loss at labeled pixels, a.k.a., partial cross-entropy loss:

$$\mathcal{L}^l_{partial} = \frac{1}{Z^{partial}} \sum_{\mathcal{P}[i] \neq 255} CE(M^l[i], \mathcal{P}[i]), \tag{8}$$

where i is the pixel index, 255 denotes unlabeled pixels which are ignored in Eq. (8) , $CE(\cdot,\cdot)$ is the cross-entropy loss function, $Z^{partial}$ is the normalizing factor, it is the number of labeled pixels.

Then we apply dense supervision to all pixels with color-prior loss [10, 40] to supplement the sparse supervision of Eq. (8). Specifically, we first compute the similarity between each pixel i and its neighboring pixel j in the LAB color space, then threshold the similarity with τ to get the affinity label $\mathcal{A}_{i,j}$, finally we supervise M as follows:

$$\mathcal{L}^l_{col} = -\frac{1}{Z^{col}} \sum_{i=1}^{HW} \sum_{j \in \mathcal{N}_i} \mathcal{A}_{i,j} \log(M^l[i]^T M^l[j]), \tag{9}$$

where $Z^{col} = \sum_{i=1}^{HW} \sum_{j \in \mathcal{N}_i} \mathcal{A}_{i,j}$ is the normalizing factor, i and j are pixel indexes, $\mathcal{N}_i$ denotes neighboring pixels of pixel i.

Besides Eqs. (8) and (9), we also supervise M^l with the pseudo label $\mathcal{M}^L$ from the last decoder layer in a self-training manner:

$$\mathcal{L}^l_{self} = \frac{1}{HW} \sum_{i=1}^{HW} CE(M^l[i], \mathcal{M}^L[i]), \tag{10}$$

where i is the pixel index. As mentioned before, pseudo labels are improved iteratively, $\mathcal{M}^L$ is more accurate than pseudo labels from earlier layers, thus $\mathcal{M}^L$ can improve earlier layers' estimation through $\mathcal{L}_{self}$.

3.4 Panoptic Branch

We adopt Panoptic Segformer [20]'s panoptic head as our panoptic branch, which contains a location decoder and a mask decoder. The location decoder aims to refine the randomly initialized thing queries by introducing the location information of different instances into them. The mask decoder predicts thing masks based on the refined thing queries and predicts stuff masks based on learned stuff queries, a classification branch is also applied on top of the thing and stuff queries to predict the class probability. The panoptic prediction is generated by merging different stuff and thing masks using a mask-wise merging strategy.

3.5 Model Training

During the training process, we train the distance branch with Eq. (4), Eq. (8), Eq. (9), and Eq. (10) as follows:

$$\mathcal{L}_{distance} = \sum_{l=1}^{L} \mathcal{L}_{col}^{l} + \mathcal{L}_{self}^{l} + \mathcal{L}_{partial}^{l} + \mathcal{L}_{cls}^{l}. \tag{11}$$

Besides, we train the panoptic branch following [20]. Specifically, we optimize the mask decoder with pseudo label M^L through dice loss, optimize the classification head with class label $\mathcal{Y}$ through focal loss, and optimize the location decoder with pseudo boxes through detection loss, the pseudo boxes are bounding boxes of different instance masks in M^L. We refer to the sum of the above losses as $\mathcal{L}_{pan}$, the total loss of our model is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{distance} + \mathcal{L}_{pan} \tag{12}$$

4 Experiments

4.1 Datasets and Evaluation Metrics

The experiments are carried out on PASCAL VOC 2012 [9] and MS COCO 2017 [23]. PASCAL VOC 2012 comprises 20 foreground classes and a background class, where the foreground classes are regarded as thing classes and the background class is treated as the stuff class. Following [10], we augment the PASCAL VOC *train* set with SBD [11], resulting in a *train_ aug* set of 10,582 images, and a *val* set of 1,449 images. MS COCO 2017 includes 80 thing classes and 53 stuff classes, and comprises 118k training images and 5k validation images. For the training labels, we adopt the single-point label $\mathcal{P}_1$ and ten-point label $\mathcal{P}_{10}$ produced by PSPS [10], which are randomly sampled from per-pixel ground truth labels with the uniform distribution. In all experiments, we employ the panoptic quality (PQ), segmentation quality (SQ), and recognition quality (RQ) to assess the model performance.

4.2 Implementation Details

Model Architecture. For the backbone, we adopt resnet50 [12] by default, we also adopt resnet101 [12] and Swin-Large [26] to get improved results. For the neck module, we adopt the deformable attention transformer [46]. For the panoptic branch, we adopt the panoptic head of Panoptic Segformer [20]. The decoder layer number L of the distance branch is 6. In general, we adopt the same backbone, neck, and panoptic branch with PSPS [10] and Point2Mask [18] for fair comparison, but any other architectures of these modules are also OK.
Model Training. Following [10, 20], We train our model with the AdamW optimizer [27], the learning rate and weight decay are 1.4e-4 and 1e-4, respectively. The mask erosion in the query aggregating process is conducted by applying erosion operation to the binarized mask 3 times with a 3×3 kernel, all the elements of this kernel are 1. More training details can be seen in Sec. B.1 of the supplementary material.

4.3 Ablation Study

In this section, we analyze the effectiveness of each component in our framework, we train the models on VOC *train_aug* set with single-point labels $\mathcal{P}_1$ and evaluate the models on VOC *val* set. The results of each analysis are described in detail below. We also provide additional ablation results in Sec. C of the supplemental material.
Analysis of Query Aggregating Feature. In our distance branch, the anchor queries are aggregated from the fusion feature F^{fuse}. Here we compare F^{fuse} with other feature maps in Tab. 1. We can see that replacing F^{fuse} with E^{learn} (learned spatial position embedding with the same size of F^{fuse}) just deteriorates model performance, this is because this learned embedding encodes less instance information than F^{fuse}. For the same reason, Replacing F^{fuse} with E^{sin} (sinusoidal spatial position embedding with the same size of F^{fuse}) also performs poorly, demonstrating aggregating anchor queries from F^{fuse} is a more efficient way to capture instance properties.

Table 1: Model performance when F^{fuse} is replaced by E^{learn} or E^{sin}.

Composition	PQ	SQ	RQ
F^{fuse}	56.6	81.4	68.1
E^{learn}	51.4	80.4	62.5
E^{sin}	51.9	80.2	63.3

Analysis of Query Enhancing. In our distance branch, after aggregating the anchor queries, we enhance them explicitly by adding class embedding E^{cls} to them through Eq. (5), and enhance them implicitly by supervising them with $\mathcal{L}_{cls}$ in Eq. (4). Here we analyse the influence of E^{cls} and $\mathcal{L}_{cls}$ in Tab. 2. The results show that better model performance can be obtained by applying E^{cls} or

$\mathcal{L}_{cls}$ alone, and applying E^{cls} and $\mathcal{L}_{cls}$ simultaneously produces the best result. It's worth noting that applying E^{cls} alone can significantly improve the result from 42.1% to 55.2%, this is because E^{cls} can enhance the query with instance class information from training labels, and help the decoder layer to discard noisy regions belonging to other classes when predicting pseudo labels.

Table 2: Model performance of query enhancing strategies. E^{cls} denotes explicit enhancing with E^{cls} through Eq. (5). $\mathcal{L}_{cls}$ denotes implicit enhancing with Eq. (4).

E^{cls}	$\mathcal{L}_{cls}$	PQ	SQ	RQ
		42.1	81.1	50.5
	✓	45.3	81.7	54.0
✓		55.2	81.6	66.3
✓	✓	56.6	81.4	68.1

Influence of Multi-scale Features. Multi-scale features are the standard setting in the panoptic segmentation field and help to segment multi-scale objects. Previous works [10, 18] adopt 3 scale features (F_1, F_2, F_3) in their models, similarly, we also adopt F_1, F_2, F_3 in our fusion module and decoder layer of the distance branch. We analyze the influence of F_1, F_2, F_3 in Tab. 3. The result shows that our model performs worse when removing features on 1 or 2 scales in the fusion module or decoder layers, demonstrating the effectiveness of these multi-scale features.

Table 3: Ablations of multi-scale features. "fusion" and "decoder" denote the fusion module and decoder layer, respectively.

fusion	F_1, F_2, F_3	F_1, F_2, F_3	F_1, F_2, F_3	F_1, F_2	F_1
decoder	F_1, F_2, F_3	F_1, F_2	F_1	F_1, F_2, F_3	F_1, F_2, F_3
PQ	56.6	55.9	55.6	56.2	55.9

Table 4: Ablations of decoder layer numbers in the distance branch.

#layers	1	2	3	4	5	6	7
PQ	3.5	27.8	54.2	55.4	55.5	56.6	55.9

Influence of Layer Numbers. Our distance branch estimates pseudo masks iteratively with 6 decoder layers. Here we train our models by adjusting the number of decoder layers in the distance branch to investigate the layer number's influence. As shown in Tab. 4, starting with layer number 1, model performance increases with the number of layers, reaches the optimal value when layer number

is 6 (our default layer number), and then declines with more layers, when layer number is too small (1 or 2), model performance is poor, this is because point masks and pseudo masks of early layers are noisy and the aggregated queries are biased, demonstrating aggregating queries with accurate pseudo masks is necessary. We also provide the visual analysis of models with different layer numbers in Sec. C.5 of the supplemental material.

Table 5: Model performance when probability maps are supervised by different losses.

Supervision	PQ	SQ	RQ
all losses	56.6	81.4	68.1
− self-training	15.6	69.6	20.4
− color-prior	46.5	76.0	59.5

Supervision for the Probability Map. As mentioned in Sec. 3.3, the probability map is trained by partial cross-entropy loss $\mathcal{L}^l_{partial}$, color-prior loss $\mathcal{L}^l_{col}$, and self-training loss $\mathcal{L}^l_{self}$. $\mathcal{L}^l_{partial}$ provides the sparse supervision from ground truth point labels. The other two losses provide dense supervision from low-level LAB information and high-level pseudo labels, respectively. Here we train our model by discarding color-prior loss or self-training loss. As shown in Tab. 5, the model performance deteriorates significantly when color-prior loss or self-training loss is removed, demonstrating that these two losses are necessary for great model performance. $\mathcal{L}^l_{partial}$ is not ablated because it provides the necessary supervision from ground truth point labels, if it is removed, the model is not supervised by any ground truth labels and can not learn to distinguish different instances.

Table 6: Results when applying different supervisions to the panoptic branch.

Supervision	PQ	SQ	RQ
point label	19.5	64.6	28.6
pseudo label	56.6	81.4	68.1

Supervision for the Panoptic Branch. The panoptic branch performs better when trained with dense pseudo labels than with point labels. To demonstrate this, we discard the distance branch and train the panoptic branch with point labels directly. For the mask decoder of the panoptic branch, we directly supervise it with point labels through partial cross-entropy loss, for the location decoder of the panoptic branch, we expand single-point labels $\mathcal{P}_1$ to 600×600 boxes to supervise it through detection loss (the model performs best with the size 600×600 when supervised by $\mathcal{P}_1$). As shown in Table 6, the panoptic branch deteriorates dramatically when supervised with point labels directly, demonstrating that training it with pseudo labels is more in line with its original de-

Table 7: Comparison with related works. $\mathcal{M}$, $\mathcal{B}$, $\mathcal{I}$ denote full mask, bounding-box, and image class label, respectively. $\mathcal{P}_1$ ($\mathcal{P}_{10}$) denotes single-point label (ten-point label).

Method	Backbone	Label	COCO			VOC		
			PQ	PQ^{th}	PQ^{st}	PQ	PQ^{th}	PQ^{st}
PanopticFCN [19]	R50	$\mathcal{M}$	43.6	49.3	35.0	67.9	66.6	92.9
Panoptic SegFormer [20]	R50	$\mathcal{M}$	48.0	52.3	41.5	69.6	68.5	92.7
Li et.al. [17]	R101	$\mathcal{B}+\mathcal{I}$	-	-	-	59.0	-	-
JTSM [35]	R18-WS [36]	$\mathcal{I}$	5.3	8.4	0.7	39.0	37.1	77.7
PSPS [10]	R50	$\mathcal{P}_1$	29.3	29.3	29.4	49.8	47.8	89.5
Point2Mask [18]	R50	$\mathcal{P}_1$	32.4	32.6	32.2	53.8	51.9	90.5
Ours	R50	$\mathcal{P}_1$	34.2	33.6	35.3	56.6	54.9	89.6
Point2Mask [18]	R101	$\mathcal{P}_1$	34.0	34.3	33.5	54.8	53.0	90.4
Ours	R101	$\mathcal{P}_1$	35.2	34.9	35.6	57.8	56.2	90.3
Point2Mask [18]	Swin-L	$\mathcal{P}_1$	37.0	37.0	36.9	61.0	59.4	93.0
Ours	Swin-L	$\mathcal{P}_1$	41.0	39.9	42.7	68.5	67.3	93.4
PanopticFCN-point [19]	R50	$\mathcal{P}_{10}$	31.2	35.7	24.3	48.0	46.2	85.2
PSPS [10]	R50	$\mathcal{P}_{10}$	33.1	33.6	32.2	56.6	54.8	91.4
Point2Mask [18]	R50	$\mathcal{P}_{10}$	35.2	36.1	34.0	59.1	57.5	91.8
Ours	R50	$\mathcal{P}_{10}$	41.3	42.6	39.3	64.0	62.6	92.1

sign for full supervision than with point labels. This is because bounding boxes from pseudo labels provide more accurate supervision for the location decoder than expanded boxes of point labels, and pseudo labels also provide more dense supervision for the mask decoder.

4.4 Comparison with Related Works

In this section, we compare our method with other related works. We train our model with $\mathcal{P}_1$ and $\mathcal{P}_{10}$ labels on MS COCO and PASCAL VOC, then evaluate our model on the *val* set of these two datasets. For fair comparison, we adopt $\mathcal{P}_1$ and $\mathcal{P}_{10}$ labels used by PSPS [10] and Point2Mask [18]. As shown in Table 7, when trained with $\mathcal{P}_1$, our method surpasses the previous SOTA model Point2Mask in all three backbone settings (R50, R101, Swin-L). When trained with $\mathcal{P}_{10}$, our model also outperforms Point2Mask by 6.1% PQ and 4.9% PQ on two datasets in R50 backbone setting. We should note that our model architecture is the same as Point2Mask except for the distance branch, the great performance improvement comes from our more accurate pseudo labels. Fig. 4 shows some pseudo label examples of our model and Point2Mask, our masks are more complete (column c), distinguish different instances with more accurate contours (column d, g, h), and contain less background noise (column a, b, e, f). Some visual examples of our predicted panoptic results are shown in Sec. D.2 of the supplemental material.

4.5 Comparison with SAM

Here we compare our model with SAM [14] by generating pseudo labels with these two models to train the same panoptic segmentation model. Specifically, we feed our $\mathcal{P}_1$ label to SAM (ViT-H [8]) and generate pseudo labels of each instance, then use these labels to train Panoptic Segformer [20] (Swin-L [26]).

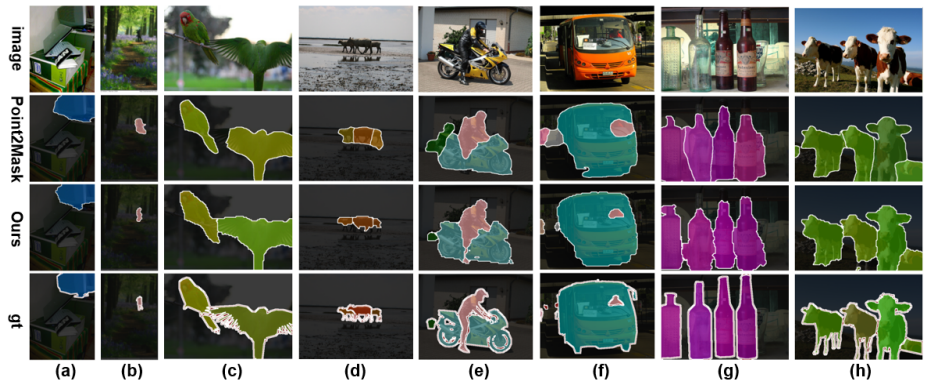

Fig. 4: Pseudo labels of Point2Mask and Ours on VOC *train* set, "gt" denotes the ground truth label. Models are based on resnet50 backbone and trained with $\mathcal{P}_1$ labels.

We also generate pseudo labels from $\mathcal{P}_1$ using the distance branch of our trained model (Swin-L) to train the Panoptic Segformer. As shown in Tab. 8, Ours (Swin-L) outperforms SAM (ViT-H) on both COCO [23] and VOC [9]. Note that, ViT-H contains more parameters than Swin-L, these results demonstrate that our distance branch trained on COCO and VOC can estimate better pseudo labels on these two datasets than SAM trained on SA-1B [14].

Table 8: Comparison with SAM. We generate pseudo labels using SAM or our trained model (trained on COCO or VOC), then train Swin-L based Panoptic Segformer with these labels. Panoptic Segformer is evaluated on the *val* set of COCO or VOC.

Pseudo Label Generation	COCO			VOC		
	PQ	PQ^{th}	PQ^{st}	PQ	PQ^{th}	PQ^{st}
from SAM (ViT-H)	46.7	51.7	39.2	60.6	62.3	26.9
from **Ours** (Swin-L)	47.3	49.0	44.7	72.2	71.1	94.1

5 Conclusion

In this paper, we propose a simple yet effective framework for point-supervised panoptic segmentation. We estimate high-quality pseudo labels based on the learnable distance rather than the rule-based distance in previous methods. Specifically, we represent each instance as an anchor query and unlabeled pixels as multi-scale features, we predict the pixel-to-instance distance according to the cross-attention between anchor queries and multi-scale features, then assign unlabeled pixels to their nearest instances to get dense pseudo labels. Extensive experiments demonstrate that our design is effective and achieves new state-of-the-art performance in point-supervised panoptic segmentation.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2022ZD0116500), the National Natural Science Foundation of China (No. U21B2042, No. 62320106010), the 2035 Innovation Program of CAS, and the InnoHK program.

References

1. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: Semantic segmentation with point supervision. In: European conference on computer vision. pp. 549–565. Springer (2016)
2. Bu, X., Peng, J., Yan, J., Tan, T., Zhang, Z.: Gaia: A transfer learning system of object detection that fits your needs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 274–283 (2021)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. pp. 213–229 (2020)
4. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1290–1299 (2022)
5. Cheng, B., Parkhi, O., Kirillov, A.: Pointly-supervised instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2617–2626 (2022)
6. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34**, 17864–17875 (2021)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
9. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge **88**, 303–338 (2009)
10. Fan, J., Zhang, Z., Tan, T.: Pointly-supervised panoptic segmentation. In: European Conference on Computer Vision. pp. 319–336. Springer (2022)
11. Hariharan, B., Arbeláez, P., Bourdev, L.D., Maji, S., Malik, J.: Semantic contours from inverse detectors. pp. 991–998 (2011)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2016)
13. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9404–9413 (2019)
14. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
15. Li, J., Fan, J., Wang, Y., Yang, Y., Zhang, Z.: Coarse mask guided interactive object segmentation. IEEE Transactions on Image Processing (2023)

16. Li, J., Fan, J., Zhang, Z.: Towards noiseless object contours for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16856–16865 (2022)
17. Li, Q., Arnab, A., Torr, P.H.: Weakly-and semi-supervised panoptic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 102–118 (2018)
18. Li, W., Yuan, Y., Wang, S., Zhu, J., Li, J., Liu, J., Zhang, L.: Point2mask: Point-supervised panoptic segmentation via optimal transport. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 572–581 (2023)
19. Li, Y., Zhao, H., Qi, X., Chen, Y., Qi, L., Wang, L., Li, Z., Sun, J., Jia, J.: Fully convolutional networks for panoptic segmentation with point-based supervision. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
20. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., Lu, T.: Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1280–1289 (2022)
21. Liang, Z., Wang, T., Zhang, X., Sun, J., Shen, J.: Tree energy loss: Towards sparsely annotated semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16907–16916 (2022)
22. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3159–3167 (2016)
23. Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. pp. 740–755 (2014)
24. Lin, Z., Zhang, Z., Chen, L.Z., Cheng, M.M., Lu, S.P.: Interactive image segmentation with first click attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13339–13348 (2020)
25. Liu, H., Peng, C., Yu, C., Wang, J., Liu, X., Yu, G., Jiang, W.: An end-to-end network for panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6172–6181 (2019)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. pp. 10012–10022 (2021)
27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proc. Int. Conf. Learning Represent. (2019)
28. Maninis, K.K., Caelles, S., Pont-Tuset, J., Gool, L.V.: Deep extreme cut: From extreme points to object segmentation. pp. 616–625 (2018)
29. Obukhov, A., Georgoulis, S., Dai, D., Van Gool, L.: Gated crf loss for weakly supervised semantic image segmentation. arXiv preprint arXiv:1906.04651 (2019)
30. Peng, J., Chang, Q., Yin, H., Bu, X., Sun, J., Xie, L., Zhang, X., Tian, Q., Zhang, Z.: Gaia-universe: Everything is super-netify. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(10), 11856–11868 (2023)
31. Peng, J., Sun, M., ZHANG, Z.X., Tan, T., Yan, J.: Efficient neural architecture transformation search in channel-level for object detection. Advances in neural information processing systems **32** (2019)
32. Peng, J., Sun, M., Zhang, Z., Tan, T., Yan, J.: Pod: Practical object detection with scale-sensitive network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9607–9616 (2019)
33. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)

34. Ruan, H., Song, H., Liu, B., Cheng, Y., Liu, Q.: Intellectual property protection for deep semantic segmentation models. Frontiers of Computer Science **17**(1), 171306 (2023)
35. Shen, Y., Cao, L., Chen, Z., Lian, F., Zhang, B., Su, C., Wu, Y., Huang, F., Ji, R.: Toward joint thing-and-stuff mining for weakly supervised panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16694–16705 (2021)
36. Shen, Y., Ji, R., Wang, Y., Chen, Z., Zheng, F., Huang, F., Wu, Y.: Enabling deep residual networks for weakly supervised object detection. In: Proc. Eur. Conf. Comp. Vis. pp. 118–136. Springer (2020)
37. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8), 888–905 (2000)
38. Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., Schroers, C.: Normalized cut loss for weakly-supervised cnn segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1818–1827 (2018)
39. Tang, M., Perazzi, F., Djelouah, A., Ben Ayed, I., Schroers, C., Boykov, Y.: On regularized losses for weakly-supervised cnn segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 507–522 (2018)
40. Tian, Z., Shen, C., Wang, X., Chen, H.: Boxinst: High-performance instance segmentation with box annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5443–5452 (2021)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
42. Wang, B., Qi, G., Tang, S., Zhang, T., Wei, Y., Li, L., Zhang, Y.: Boundary perception guidance: A scribble-supervised semantic segmentation approach. In: IJCAI International joint conference on artificial intelligence (2019)
43. Zhang, S., Liew, J.H., Wei, Y., Wei, S., Zhao, Y.: Interactive object segmentation with inside-outside guidance. pp. 12234–12244 (2020)
44. Zhang, W., Pang, J., Chen, K., Loy, C.C.: K-net: Towards unified image segmentation. Advances in Neural Information Processing Systems **34**, 10326–10338 (2021)
45. Zhang, Z., Pan, C., Peng, J.: Delving into the effectiveness of receptive fields: Learning scale-transferrable architectures for practical object detection. International Journal of Computer Vision **130**(4), 970–989 (2022)
46. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)