Fully Data-Driven Pseudo Label Estimation for Pointly-Supervised Panoptic Segmentation

Jing Li[1,2,4], **Junsong Fan**[3], **Yuran Yang**[5], **Shuqi Mei**[5], **Jun Xiao**[1], **Zhaoxiang Zhang**[1,2,3,4*]

[1]University of Chinese Academy of Sciences (UCAS)
[2]Institute of Automation, Chinese Academy of Sciences (CASIA)
[3]Centre for Artificial Intelligence and Robotics, HKISI_CAS
[4]State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS)
[5]Tencent Maps, Tencent
{lijing2018, junsong.fan}@ia.ac.cn, {yuranyang, shawnmei}@tencent.com, xiaojun@ucas.ac.cn, zhaoxiang.zhang@ia.ac.cn

Abstract

The core of pointly-supervised panoptic segmentation is estimating accurate dense pseudo labels from sparse point labels to train the panoptic head. Previous works generate pseudo labels mainly based on hand-crafted rules, such as connecting multiple points into polygon masks, or assigning the label information of labeled pixels to unlabeled pixels based on the artificially defined traversing distance. The accuracy of pseudo labels is limited by the quality of the hand-crafted rules (polygon masks are rough at object contour regions, and the traversing distance error will result in wrong pseudo labels). To overcome the limitation of hand-crafted rules, we estimate pseudo labels with a fully data-driven pseudo label branch, which is optimized by point labels end-to-end and predicts more accurate pseudo labels than previous methods. We also train an auxiliary semantic branch with point labels, it assists the training of the pseudo label branch by transferring semantic segmentation knowledge through shared parameters. Experiments on Pascal VOC and MS COCO demonstrate that our approach is effective and shows state-of-the-art performance compared with related works. Codes are available at https://github.com/BraveGroup/FDD.

Introduction

Panoptic segmentation is a computer vision task that partitions an image into non-overlapping masks for both thing objects and stuff categories (Kirillov et al. 2019b). With the development of neural network technology (Zhiqiang Chen 2022; Mengya Han 2023; Jianing Han 2023; Jiaqi Li 2023; Qi Zheng 2023; Guyue Hu 2023; Cheng-Cheng Ma 2023), deep learning-based panoptic segmentation models have shown promising performance, but their effectiveness relies heavily on pixel-wise training labels, and annotating these labels is time-consuming. The high annotation cost hinders the widespread use of these methods in practical applications.

To reduce the heavy annotation burden, some works (Li, Arnab, and Torr 2018; Shen et al. 2021; Li et al. 2022a; Fan, Zhang, and Tan 2022) propose to train panoptic segmentation models with pixel-wise pseudo labels generated from image tags, bounding boxes, and point labels. In these works, image-tag based model (Shen et al. 2021) performs much worse than those based on point or bounding box labels and is still far from practical use. Box labels (Li, Arnab, and Torr 2018) cannot be flexibly applied to uncountable stuff categories whose shapes are complex and various. Point labels (Li et al. 2022a; Fan, Zhang, and Tan 2022) apply to both thing and stuff categories, and their annotation cost can be adjusted by changing point numbers in different datasets and tasks, thus attracting the most attention in recent studies. In this paper, we focus on boosting the performance of pointly-supervised panoptic segmentation model.

Since point labels are sparse, which means the size and shape information of instances are missing, previous methods usually generate dense pseudo labels from point labels

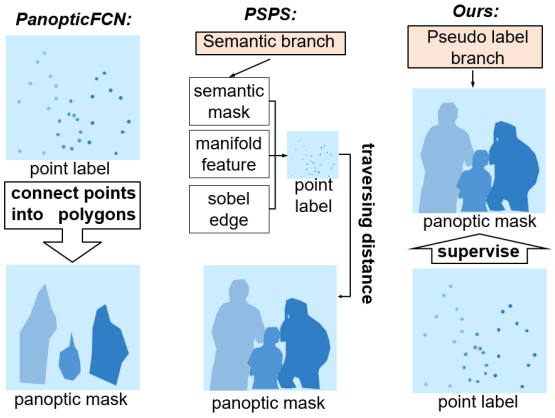

Figure 1: Pseudo label estimation process comparison with PanopticFCN(Li et al. 2022a) and PSPS(Fan, Zhang, and Tan 2022). Both PanopticFCN and PSPS are based on hand-crafted rules (connecting points into polygon masks, assigning labeled points to unlabeled pixels based on the artificially defined traversing distance), their pseudo label estimations are not supervised by point labels. Our model is based on a fully data-driven pseudo label branch, this branch is optimized by point labels end-to-end, our pseudo label estimations are supervised by point labels.

*Corresponding author

based on hand-crafted rules, the quality of hand-crafted rules limits the accuracy of pseudo labels. As shown in Figure 1, PanopticFCN (Li et al. 2022a) generates the dense pseudo labels by simply connecting point labels into polygon masks, these polygon masks lack detail results at contour regions where most hard example pixels with large ambiguity locate. PSPS (Fan, Zhang, and Tan 2022) first trains a semantic branch to generate dense semantic segmentation masks, then assigns the label information of labeled points to appropriate unlabeled pixels based on the artificially defined traversing distance, the traversing distance is based on semantic segmentation masks, manifold features, and Sobel edges, the errors in these three results will cause the traversing distance error and deteriorate pseudo labels' quality.

To overcome the limitation of hand-crafted rules, we utilize a fully data-driven pseudo label branch to estimate dense pseudo labels. As shown in Figure 1, the pseudo label branch predicts dense pseudo labels directly and is supervised by point labels end-to-end, it estimates pseudo labels by learning from all point labels in the whole dataset rather than based on artificially defined rules, thus our pseudo labels are more accurate than previous methods.

Our pseudo label branch is specifically designed for pseudo label estimation in two aspects. Firstly, the number of ground truth (GT) instances N varies for different input images, our pseudo label branch adaptively estimates different numbers of instance probability maps for different images. To achieve this goal, our model generates N instance-aware queries based on N GT instances of the point label, each query encodes the location and instance information of one instance. The pseudo label branch predicts N probability maps according to these N queries, respectively, each probability map is optimized to highlight the corresponding GT instance region. Secondly, we also train an auxiliary semantic branch with point labels, this branch assists the training of the pseudo label branch by transferring semantic segmentation knowledge through shared parameters.

The contributions of this work can be summarized as:

- We propose a fully data-driven module to estimate dense pseudo labels from point labels for pointly-supervised panoptic segmentation.
- We design an auxiliary branch training strategy that transfers semantic segmentation knowledge to the pseudo label branch through shared parameters.
- We conduct experiments to demonstrate the effectiveness of our method and get the new state-of-the-art performance with 63.1% PQ on VOC and 40.3% PQ on COCO.

Related Works

Panoptic Segmentation

Panoptic segmentation (Kirillov et al. 2019b) task combines semantic segmentation and instance segmentation together, it aims to assign a semantic class label and an instance ID label to each pixel. (Kirillov et al. 2019b) solves this problem by directly combing semantic segmentation and instance segmentation results, then (Kirillov et al. 2019a) improves the results by adopting a shared Feature Pyramid Network (FPN) backbone. OANet (Liu et al. 2019) utilizes a spatial ranking module to deal with the occlusion problem between different thing instances. UPSNet (Xiong et al. 2019) resolves the conflicts between semantic and instance segmentation via pixel-wise classification.

Recently, transformer-based detection models DETR (Carion et al. 2020) and DeformableDETR (Zhu et al. 2020) have shown great success, many transformer-based panoptic segmentation models have been proposed after them. Panoptic SegFormer (Li et al. 2022b) adopts two separate query sets to represent thing and stuff contents. Mask2Former (Cheng et al. 2022) constrains cross-attention within predicted mask regions to extract localized features.

Weakly Supervised Panoptic Segmentation

To reduce the annotation burden of panoptic masks, some works adopt weak annotations to train models, including image tags, bounding boxes, and points. (Li, Arnab, and Torr 2018) supervises stuff regions with image tags and supervises instance regions with bounding boxes. JTSM (Shen et al. 2021) generates pseudo labels for thing and stuff targets only using image tags. PSIS (Cheng, Parkhi, and Kirillov 2022) supervises instance segmentation models with point labels sampled in ground truth boxes. (Shen et al. 2019) solves the panoptic segmentation problem by utilizing scribbles as guidance for mask prediction. PanopticFCN (Li et al. 2022a) assigns multiple point labels to one target and connects them into polygon masks. PSPS (Fan, Zhang, and Tan 2022) assigns the label information of labeled points to unlabeled pixels based on the traversing distance.

Sparsely Supervised Semantic Segmentation

To reduce the annotation burden in semantic segmentation, many works adopt sparse labels to train models, including point(Liang et al. 2022; Obukhov et al. 2019; Bearman et al. 2016), scribble (Lin et al. 2016; Tang et al. 2018a; Shi and Malik 2000; Tang et al. 2018b; Obukhov et al. 2019; Liang et al. 2022; Wang et al. 2019) and block(Liang et al. 2022) labels. These methods usually adopt sparse labels to supervise segmentation results with partial cross entropy loss and apply color-based loss (CRF loss, Gated CRF, etc.) to all pixels to supplement the sparse label supervision.

Approach

To bridge the gap between point labels and dense pixel-wise labels in pointly-supervised panoptic segmentation task, we estimate dense pseudo labels from point labels as the training label. For an image of size $H \times W$ containing N instances (we take one thing object or one stuff category as one instance in this paper), the point panoptic label offers the semantic class label and instance ID label at several pixels (instance ID label is also applied to stuff regions in this paper), thus we can get the sparse point semantic label $S^{sem} \in \{1, 2, \cdots, C, 255\}^{H \times W}$, the sparse point instance label $S^{ins} \in \{1, 2, \cdots, N, 255\}^{H \times W}$, and N instances' class labels $Y \in \{1, 2, \cdots, C\}^{N}$, here C is the number of semantic classes, 255 denotes unlabeled pixels, the right part of Figure 2 illustrates S^{ins} and S^{sem}. In our paper, we

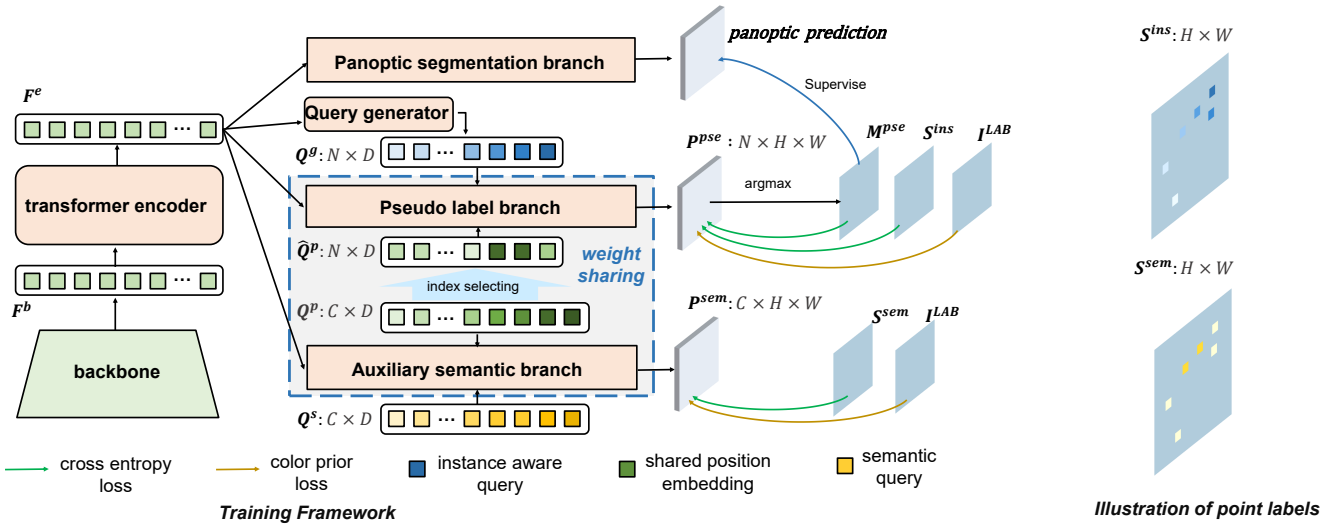

Figure 2: Illustration of the training framework of point labels. The transformer encoder refines the backbone features F^b as F^e. The query generator aggregates F^e at labeled regions of point instance label S^{ins} and produces instance aware queries Q^g. The pseudo label branch takes Q^g and F^e as input to estimate dense pseudo instance label M^{pse} to supervise the panoptic segmentation branch, it is supervised by S^{ins}, M^{pse} and LAB format image I^{LAB}. The auxiliary semantic branch shares the layer weight and position embedding with the pseudo label branch, it is supervised by point semantic label S^{sem} and I^{LAB}. During model testing, the query generator, pseudo label branch, and auxiliary semantic branch are removed, panoptic segmentation branch is adopted to predict segmentation results.

first estimate the dense pseudo instance label M^{pse} from point labels with a fully data-driven module, then supervise the panoptic head with M^{pse} and class labels Y (the dense pseudo panoptic label can be inferred from M^{pse} and Y).

In the following, we will elaborate on the overall framework and each key module of our framework.

Overall Framework

As shown in Figure 2, the backbone encodes input images as backbone features F^b, the transformer encoder encodes F^b as feature tokens F^e, F^e is fed into four key modules, namely query generator, pseudo label branch, auxiliary semantic branch, and panoptic segmentation branch.

The query generator generates instance-aware queries Q^g that are fed into the pseudo label branch to guide the pseudo label estimation.

The pseudo label branch is a fully data-driven module and is supervised by point instance label S^{ins} end-to-end. This branch adopts a transformer-based architecture that takes queries and position embeddings as input, it generates pseudo instance label M^{pse} based on instance-aware queries Q^g to train the panoptic segmentation branch.

The auxiliary semantic branch adopts the same architecture as the pseudo label branch, it assists the training of the pseudo label branch by transferring semantic segmentation knowledge through the shared layer weight and position embedding.

During model testing, the query generator, pseudo label branch, and auxiliary semantic branch are all removed and the trained panoptic branch is adopted to predict panoptic segmentation results.

Query Generator

The query generator generates instance-aware queries based on feature tokens F^e and point instance label S^{ins}. Specifically, the 1D feature tokens F^e are first reshaped into 2D feature maps of size $H/8 \times W/8$, then the query generator projects the feature maps with two convolution layers and gets $F^{proj} \in \mathbb{R}^{D \times H/8 \times W/8}$, D is the feature channel number, then we apply dilatation morphology to S^{ins} to expand the labeled regions, and downsample the expanded result to get $\hat{S}^{ins} \in \{1, 2, \cdots, N, 255\}^{H/8 \times W/8}$, the instance-aware queries $Q^g \in \mathbb{R}^{N \times D}$ of N instances are generated as follows:

$$Q^g[n] = \max_{\{i|\hat{S}^{ins}[i]=n\}} F^{proj}[i], \tag{1}$$

where i is the pixel index, n denotes the n-th query of Q^g, the max operation is conducted for each feature channel of $F^{proj}[i] \in \mathbb{R}^D$ independently. Eq. 1 aggregates the features at pixels labeled as n through max operation, thus $Q^g[n]$ encodes the location and instance information of the n-th instance, it can guide this instance's mask estimation.

Pseudo Label Branch

Similar to the panoptic head of Panoptic Segformer (Li et al. 2022b), our pseudo label branch contains several stacked transformer decoder layers, it takes several queries and corresponding position embeddings as input and predicts a single-channel probability map for each query, the query is refined every time it passes through a transformer decoder layer, while the position embedding is not refined.

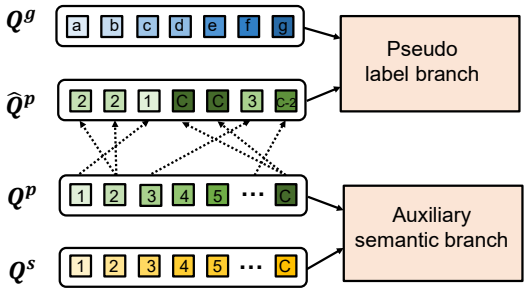

Figure 3: Index selecting position embeddings. "a, b, ..., g" are instance ID labels, "1, 2, ..., C" are semantic class labels. In this figure, instance class labels are "2, 2, 1, C, C, 3, C-2".

As shown in Figure 2, for N labeled instances in the input image, N instance-aware queries Q^g are generated by the query generator, and N position embeddings $\hat{Q}^p$ are index selected from C shared semantic position embeddings utilizing N instances' class labels Y, the selecting process is shown in Figure 3. Then the pseudo label branch predicts N single-channel probability maps based on Q^g and $\hat{Q}^p$, each single-channel probability map is correlated with one instance and highlights the instance region in the input image. These single-channel probability maps form a N-channel probability map $P^{pse} \in \mathbb{R}^{N\times H\times W}$, and the pseudo instance label M^{pse} is estimated as follows:

$$M^{pse}[i] = \arg\max_n P^{pse}_n[i], \tag{2}$$

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

To train the pseudo label branch, we adopt the point instance label S^{ins} to optimize P^{pse} with partial cross-entropy loss $\mathcal{L}_{pCE}$, and apply the color-prior loss (Fan, Zhang, and Tan 2022; Tian et al. 2021) $\mathcal{L}_{col}$ to all pixels of P^{pse} to supplement the sparse supervision of S^{ins}.

The partial cross entropy loss $\mathcal{L}_{pCE}$ is defined as

$$\mathcal{L}_{pCE}(P^{pse}, S^{ins}) = -\frac{1}{Z^{pCE}} \sum_{S^{ins}[i]\neq 255} \log P^{pse}_{S^{ins}[i]}[i], \tag{3}$$

where i is the pixel index, $S^{ins}[i]$ is the instance ID label of i-th pixel, $P^{pse}_{S^{ins}[i]}[i]$ is the $S^{ins}[i]$-th channel of $P^{pse}[i]$, 255 denotes unlabeled pixels which are ignored in Eq. 3, Z^{pCE} is the normalizing factor, it is the number of labeled pixels.

The color-prior loss $\mathcal{L}_{col}$ is defined as:

$$\mathcal{L}_{col} = -\frac{1}{Z^{col}} \sum_{i=1}^{HW} \sum_{j\in\mathcal{N}_i} A_{i,j} \log P^{pse}[i]^T P^{pse}[j], \tag{4}$$

where $Z^{col} = \sum_{i=1}^{HW}\sum_{j\in\mathcal{N}_i} A_{i,j}$ is the normalizing factor, i and j are pixel indexes, $\mathcal{N}_i$ denotes neighboring pixels of i, $A_{i,j}$ is the color affinity and is computed as follows:

$$A_{i,j} = \begin{cases} 1 & \text{if } exp\{-\frac{1}{2}\|I^{LAB}[i] - I^{LAB}[j]\|_2\} > 0.3 \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where I^{LAB} is the LAB color format of input image I, $\|\cdot\|_2$ is the two norm function.

Besides Eq. 4 and 5, we further supervise P^{pse} with the pseudo instance label M^{pse} in a self-training manner through cross-entropy loss $\mathcal{L}_{CE}$:

$$\mathcal{L}_{CE}(P^{pse}, M^{pse}) = -\frac{1}{HW}\sum_{i=1}^{HW} \log P^{pse}_{M^{pse}[i]}[i], \tag{6}$$

where i is the pixel index, $M^{pse}[i] = \arg\max_n P^{pse}_n[i]$ indicates the most confident channel of $P^{pse}[i]$, this loss optimizes the most confident channel $P^{pse}_{n_i}[i]$ towards 1 and reduces the uncertainty of $P^{pse}[i]$. Since most pixels in M^{pse} contain right labels, this loss optimizes P^{pse}'s most pixels in the right direction and improves P^{pse}'s quality.

With the constraints of S^{ins}, $\mathcal{L}_{col}$, and M^{pse}, the total loss for the pseudo label branch is:

$$\mathcal{L}_{pse} = \mathcal{L}_{pCE}(P^{pse}, S^{ins}) + \mathcal{L}_{CE}(P^{pse}, M^{pse}) + \mathcal{L}_{col} \tag{7}$$

Auxiliary Semantic Branch

The auxiliary semantic branch adopts the same architecture as the pseudo label branch and shares the layer weight and position embeddings with the latter. As shown in Figure 2, this branch takes C semantic queries and semantic position embeddings as input and predicts a C-channel semantic segmentation map $P^{sem} \in \mathbb{R}^{C\times H\times W}$, the c-th channel highlights the region belonging to the c-th semantic class.

Similar to the pseudo label branch, we adopt point semantic label S^{sem} and color-prior loss to supervise P^{sem}. The color-prior loss for P^{sem} is

$$\mathcal{L}^{sem}_{col} = -\frac{1}{Z^{col}} \sum_{i=1}^{HW} \sum_{j\in\mathcal{N}_i} A_{i,j} \log(P^{sem}[i]^T P^{sem}[j]), \tag{8}$$

where Z^{col}, $\mathcal{N}_i$, and $A_{i,j}$ are the same as those in Eq. 4 of $\mathcal{L}_{col}$. The total loss for P^{sem} is as follows:

$$\mathcal{L}_{sem} = \mathcal{L}_{pCE}(P^{sem}, S^{sem}) + \mathcal{L}^{sem}_{col}, \tag{9}$$

where partial cross entropy loss $\mathcal{L}_{pCE}(P^{sem}, S^{sem})$ supervises P^{sem} with labeled pixels in S^{sem} through cross entropy loss and ignores unlabeled pixels. Here we don't utilize the dense pseudo semantic label M^{sem} generated from P^{sem} because auxiliary semantic branch aims to assist the pseudo label branch training, M^{sem} may improve auxiliary semantic branch's performance but also brings more constraints to the shared parameters of pseudo label branch and deteriorates the quality of M^{pse}.

Panoptic Segmentation Branch

We adopt Panoptic Segformer (Li et al. 2022b)'s panoptic head as our panoptic segmentation branch, which contains a location decoder, a mask decoder, and a classification branch, these three modules are trained with detection loss, dice loss, and focal loss, respectively, the detail of these losses is in (Li et al. 2022b), here we simply adopt $\mathcal{L}_{pan}$ to denote the sum of these losses. we adopt the pseudo instance label M^{pse} and class label Y mentioned above to optimize the dice loss and focal loss, and generate bounding boxes from M^{pse} to optimize the detection loss.

Model Training

During the training process, the pseudo label branch, auxiliary semantic branch, and panoptic segmentation branch are optimized simultaneously by loss $\mathcal{L}_{pse}$, $\mathcal{L}_{sem}$ and $\mathcal{L}_{pan}$, respectively, the total loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{pse} + \lambda_{sem}\mathcal{L}_{sem} + \mathcal{L}_{pan}, \tag{10}$$

where λ_{sem} is a balance factor to adjust the influence of the auxiliary semantic branch on the pseudo label branch.

Experiments

Datasets and Evaluation Metrics

All experiments are carried out on PASCAL VOC 2012 (Everingham et al. 2009) and MS COCO 2017 (Lin et al. 2014). VOC comprises 20 thing classes (foreground classes) and a stuff class (background class). Following (Fan, Zhang, and Tan 2022), we augment the VOC *train* set with SBD *train* set (Hariharan et al. 2011), getting a training set of 10,582 images, and refer to this set as *train_aug* set. COCO includes 80 thing classes and 53 stuff classes, it comprises 118,000 training images and 5,000 validation images. We employ the panoptic quality (PQ) metric to assess the segmentation performance of trained models.

Since there are no publicly available manually annotated point labels for panoptic segmentation on COCO and VOC, previous works sample point labels from ground truth masks to simulate the human annotating process. We follow PSPS (Fan, Zhang, and Tan 2022) and generate point labels by randomly sampling points from the ground truth masks with a uniform distribution. In our paper, We adopt single-point label $\mathcal{P}_1$ (one point label per instance) and ten-point label $\mathcal{P}_{10}$ (ten point labels per instance). The point label is sampled once and fixed in all experiments.

Implementation Details

We build our framework based on Panoptic SegFormer(Li et al. 2022b) with a resnet50 (He et al. 2016) backbone by adding a pseudo label branch, an auxiliary semantic branch, and a query generator to it, the panoptic segmentation branch is the same as that of Panoptic SegFormer. Our model adopts the same training recipe of (Fan, Zhang, and Tan 2022), namely AdamW optimizer with weight decay 1e-4 and learning rate 1.4e-4. Besides, we apply a linear warm-up schedule to the losses utilizing M^{pse} as supervision to reduce the influence of noisy M^{pse} at early training epochs. The color-prior loss adopts the same setting in (Fan, Zhang, and Tan 2022). λ_{sem} is set to 1 and 0.1 for $\mathcal{P}_1$ and $\mathcal{P}_{10}$ settings, respectively.

Ablation Study

In this part, we conduct experiments to analyze the effectiveness of each module in our framework, all the models are trained on VOC *train_aug* set and evaluated on VOC *val* set with PQ. We adopt single-point label $\mathcal{P}_1$ and ten-point label $\mathcal{P}_{10}$ as supervision. Specifically, we analyze the influence of balance factor λ_{sem}, parameter sharing strategy, self-training, color-prior loss, point sampling strategy, and supervision for the panoptic branch on model performance. The following is a detailed description of these analyses. We also show additional analysis results in the supplemental material.

λ_{sem}	$\mathcal{P}_1$	$\mathcal{P}_{10}$
1.5	50.4	61.9
1	52.0	61.4
0.5	51.9	62.9
0.1	46.8	63.1
0	32.9	61.2

Table 1: Influence of balance factor λ_{sem}.

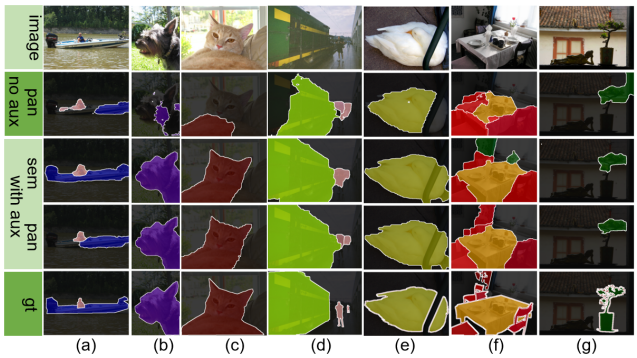

Figure 4: "with aux" denotes "with auxiliary branch", "no aux" denotes "without auxiliary branch ($\lambda_{sem} = 0$)", "sem" denotes semantic segmentation results from auxiliary branch, "pan" denotes pseudo panoptic labels from pseudo label branch, models are trained with $\mathcal{P}_1$. The model without auxiliary branch usually underestimates ((a), (b), (c), (d), (e)) or overestimates ((f), (g)) the foreground object region.

Influence of Balance Factor λ_{sem}. The balance factor λ_{sem} in Eq. 10 determines the optimization priority of the auxiliary semantic branch. We train the model by setting λ_{sem} to different values and the results are shown in Table 1, the model gets the best performance with $\lambda_{sem} = 1$ and $\lambda_{sem} = 0.1$ when trained with $\mathcal{P}_1$ and $\mathcal{P}_{10}$, respectively. Models with $\lambda_{sem} > 0$ always perform better than those with $\lambda_{sem} = 0$ in both $\mathcal{P}_1$ and $\mathcal{P}_{10}$ settings. We also compare pseudo labels of the model with or without ($\lambda_{sem} = 0$) auxiliary branch in Figure 4, the model without auxiliary branch usually underestimates or overestimates the foreground object region. These results demonstrate that the constraints of the auxiliary semantic branch play a critical role in assisting the pseudo label branch training.

Influence of Parameter Sharing Strategy. Our auxiliary semantic branch transfers semantic segmentation knowledge to pseudo label branch by sharing its layer weight and position embedding. Here we also evaluate model's performance when using independent layer weight or position embedding. The result in Table 2 shows that sharing layer weight and position embedding performs best, demonstrating our default sharing strategy is the best.

Instance Information Injection Choice. Our model injects

layer	pos	$\mathcal{P}_1$	$\mathcal{P}_{10}$
		49.7	62.4
	✓	49.9	62.3
✓		50.3	62.7
✓	✓	52.0	63.1

Table 2: Influence of parameter sharing strategy. "layer, pos" denote parameter sharing of layer weight and position embedding, respectively.

Settings	$\mathcal{P}_1$	$\mathcal{P}_{10}$
pos	14.2	62.7
query	52.0	63.1

Table 3: Ablation results of instance information injection choice . " query, pos" denote injecting instance information to the query and position embedding of the pseudo label branch, respectively.

instance information to pseudo label branch by generating instance-aware queries for it, here we also generate instance-aware position embeddings for it with the query generator and feed the shared semantic queries to it. As shown in Table 3, injecting instance information to position embeddings performs much worse, demonstrating the effectiveness of our default injection choice.

Generate Q^g from	$\mathcal{P}_1$	$\mathcal{P}_{10}$
F^{proj}	52.0	63.1
2D embedding	43.7	61.0

Table 4: Query feature map ablation results in PQ.

Query Feature Map Choice. The query generator generates instance-aware query Q^g from the feature map F^{proj}, here we replace F^{proj} with a directly learned 2D embedding of the same size to generate Q^g. As shown in Table 4, the new model performs worse than our default model, this is because the learned embedding encodes less instance information than F^{proj} and fails to guide pseudo label branch to estimate accurate pseudo labels.

auxiliary	pseudo	$\mathcal{P}_1$	$\mathcal{P}_{10}$
		50.0	62.5
	✓	52.0	63.1
✓		48.5	62.4
✓	✓	50.9	62.5

Table 5: Ablation results of self-training.

Influence of Self-Training. In this part, we conduct experiments to study the influence of self-training. We supervise the pseudo label branch with M^{pse} through Eq. 6 in a self-training manner by default, here we also get M^{sem} from mask logits P^{sem} of the auxiliary semantic branch and supervise P^{sem} with M^{sem} in a self-training manner. The results when applying self-training to different branches are shown in Table 5. Our model performs best in both $\mathcal{P}_1$ and $\mathcal{P}_{10}$ settings when applying self-training to the pseudo label branch alone, further adding self-training loss to the auxiliary semantic branch just deteriorates model's performance. We think this is because self-training for the auxiliary semantic branch forces the shared layer weight to learn to predict better semantic segmentation results and hinders the pseudo label branch's instance discrimination learning.

auxiliary	pseudo	$\mathcal{P}_1$	$\mathcal{P}_{10}$
		30.3	49.8
	✓	44.3	62.4
✓		34.9	52.8
✓	✓	52.0	63.1

Table 6: Results when applying $\mathcal{L}_{col}$ to different branches.

Influence of Color-Prior Loss. In this part, we conduct experiments to study the influence of color prior loss on the pseudo label branch and auxiliary semantic branch. We train the model by applying this loss to different branches and the results are shown in Table 6. The model performs best when applying this loss to both branches and deteriorates when removing this loss. This loss supplements the sparse supervision of point labels by applying dense color-based constraints to all pixels, thus improving the model's performance.

Strategy	$\mathcal{P}_1$	$\mathcal{P}_{10}$
center-biased	53.0	62.4
uniform	52.0	63.1
border-biased	44.8	60.2

Table 7: Ablation results of point sampling strategy.

Influence of Point Sampling Strategy. In this part, we study the influence of point label sampling strategies. By default, we randomly sample point labels from ground truth masks with a uniform distribution. Following PSPS, we also sample points with border-biased strategy and center-biased strategy. The border-biased strategy first builds a probability density map according to the square of Euclidean distance from each pixel to the centroid of the corresponding instance mask, then samples points based on the normalized probability map. The center-biased strategy samples points similarly by reversing the distance based probability map. As shown in Table 7, the center-biased strategy and uniform strategy perform similarly, the border-biased strategy performs much worse than the other two strategies. We should note that the center-biased strategy and uniform strategy are more in line with human intuition and easier to conduct than the border-biased strategy, thus our model will perform well with manually annotated points in practice.

Supervision	$\mathcal{P}_1$	$\mathcal{P}_{10}$
point label	22.6	56.5
pseudo label	52.0	63.1

Table 8: Supervision ablation for the panoptic branch.

Method	Backbone	Label	COCO			VOC		
			PQ	PQ^{th}	PQ^{st}	PQ	PQ^{th}	PQ^{st}
PanopticFCN (Li et al. 2022a)	R50	$\mathcal{M}$	43.6	49.3	35.0	67.9	66.6	92.9
Panoptic SegFormer (Li et al. 2022b)	R50	$\mathcal{M}$	48.0	52.3	41.5	69.6	68.5	92.7
Li et.al. (Li, Arnab, and Torr 2018)	R101	$\mathcal{B}+\mathcal{I}$	-	-	-	59.0	-	-
Combination (Ahn and Kwak 2018; Ahn, Cho, and Kwak 2019)	R50	$\mathcal{I}$	-	-	-	37.1	35.5	74.2
JTSM (Shen et al. 2021)	R18-WS	$\mathcal{I}$	5.3	8.4	0.7	39.0	37.1	77.7
PanopticFCN-point (Li et al. 2022a)	R50	$\mathcal{P}_{10}$	31.2	35.7	24.3	48.0	46.2	85.2
PSPS (Fan, Zhang, and Tan 2022)	R50	$\mathcal{P}_1$	29.3	29.3	29.4	49.8	47.8	89.5
PSPS (Fan, Zhang, and Tan 2022)	R50	$\mathcal{P}_{10}$	33.1	33.6	32.2	56.6	54.8	91.4
Ours	R50	$\mathcal{P}_1$	33.0	31.8	34.9	52.0	50.1	89.0
Ours	R50	$\mathcal{P}_{10}$	40.3	41.4	38.6	63.1	61.7	91.8

Table 9: Comparison with other SOTA works on VOC and COCO datasets. All the models are evaluated on MS COCO *val* set and VOC *val* set with PQ, PQ^{th}, and PQ^{st}. $\mathcal{M}$ denotes full mask supervision, $\mathcal{B}$ denotes bounding-box supervision, $\mathcal{I}$ denotes image class label supervision, $\mathcal{P}_1$ ($\mathcal{P}_{10}$) denotes supervision with one point label (ten point labels) per instance.

Supervision for the Panoptic Branch. By default, we train the panoptic branch with dense pseudo instance labels M^{pse} from the pseudo label branch. In this part, we keep the pseudo label branch and the auxiliary semantic branch intact and train the panoptic branch with point labels. Specifically, we supervise the panoptic branch's mask decoder with point labels through partial cross entropy loss, for the location decoder of the panoptic branch, we use bounding boxes of $\mathcal{P}_{10}$ labels and expand $\mathcal{P}_1$ labels to 400×400 boxes to train it through detection loss (the model performs best with the size 400×400 in $\mathcal{P}_1$ setting). As shown in Table 8, the panoptic branch deteriorates dramatically when supervised with point labels directly, demonstrating that estimating dense pseudo labels from point labels to train the panoptic head is necessary. This is because bounding boxes from dense pseudo labels are more accurate than those from point labels (especially for one-point labels), and pseudo labels provide more dense supervision than point labels, the location decoder and mask decoder of the panoptic branch can be optimized better with pseudo labels.

Comparison with Related Works

In this part, 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, $\mathcal{P}_1$ and $\mathcal{P}_{10}$ are generated in the same way as PSPS. As shown in Table 9, in the $\mathcal{P}_{10}$ setting, our method improves previous SOTA model PSPS by 7.2% and 6.5% on two datasets. In the $\mathcal{P}_1$ setting, our model also outperforms PSPS by 3.7% and 2.2% on two datasets, showing comparable performance (33.0% vs 33.1%) with PSPS trained by $\mathcal{P}_{10}$ on MS COCO. We should note that both our method and PSPS are based on Panoptic SegFormer, the great boost of model performance comes from the improvement of pseudo labels. We compare our pseudo labels with PSPS's in Figure 5, PSPS's hand-crafted traversing distance is directly based on its semantic segmentation results, semantic segmentation errors deteriorate the pseudo panoptic label, our auxiliary semantic branch just assists the model training and semantic segmentation errors don't influence our panoptic label, our pseudo labels are more accurate than those of PSPS.

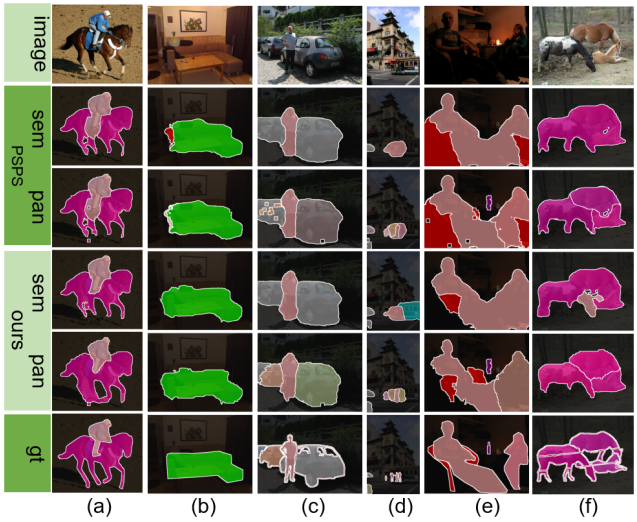

Figure 5: Pseudo label comparison with PSPS, "sem" denotes semantic segmentation results from semantic branch of PSPS or our auxiliary branch, "pan" denotes the pseudo panoptic label. Semantic segmentation errors are introduced to panoptic labels in PSPS model ((a), (b), (c), (e)), these errors don't influence the panoptic label in our model ((c), (d), (e), (f)). Models are trained with $\mathcal{P}_{10}$ label.

Conclusion

In this paper, we propose a fully data-driven pseudo label branch to estimate dense pseudo labels from point labels to train panoptic segmentation models. This branch is optimized by point labels end-to-end, it learns from all point labels of the dataset and estimates more accurate pseudo labels than previous methods which estimate pseudo labels based on hand-crafted rules. Besides, we adopt an auxiliary branch to assist the training of the pseudo label branch by sharing parameters. Experiments demonstrate the effectiveness of our method and our model achieves new state-of-the-art performance.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231); and in part by the InnoHK Program.

References

Ahn, J.; Cho, S.; and Kwak, S. 2019. Weakly Supervised Learning of Instance Segmentation With Inter-Pixel Relations.

Ahn, J.; and Kwak, S. 2018. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*.

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, 549–565. Springer.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. 213–229.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1290–1299.

Cheng, B.; Parkhi, O.; and Kirillov, A. 2022. Pointly-supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2617–2626.

Cheng-Cheng Ma, Y.-B. F. Y. Z. Z.-F. L., Bao-Yuan Wu. 2023. Effective and Robust Detection of Adversarial Examples via Benford-Fourier Coefficients. *Machine Intelligence Research*, 20: 666–682.

Everingham, M.; Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2009. The Pascal Visual Object Classes (VOC) Challenge. 88: 303–338.

Fan, J.; Zhang, Z.; and Tan, T. 2022. Pointly-Supervised Panoptic Segmentation. In *European Conference on Computer Vision*, 319–336. Springer.

Guyue Hu, H. Z., Bin He. 2023. Compositional Prompting Video-language Models to Understand Procedure in Instructional Videos. *Machine Intelligence Research*, 20: 249–262.

Hariharan, B.; Arbeláez, P.; Bourdev, L. D.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. 991–998.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition.

Jianing Han, J. S. H. T., Ziming Wang. 2023. Symmetric-threshold ReLU for Fast and Nearly Lossless ANN-SNN Conversion. *Machine Intelligence Research*, 20: 435–446.

Jiaqi Li, L. W. X. Z., Zhuofeng Li. 2023. Machine Learning in Lung Cancer Radiomics. *Machine Intelligence Research*, 20: 753–782.

Kirillov, A.; Girshick, R.; He, K.; and Dollár, P. 2019a. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6399–6408.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019b. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9404–9413.

Li, Q.; Arnab, A.; and Torr, P. H. 2018. Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 102–118.

Li, Y.; Zhao, H.; Qi, X.; Chen, Y.; Qi, L.; Wang, L.; Li, Z.; Sun, J.; and Jia, J. 2022a. Fully convolutional networks for panoptic segmentation with point-based supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, Z.; Wang, W.; Xie, E.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; Luo, P.; and Lu, T. 2022b. Panoptic SegFormer: Delving deeper into panoptic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1280–1289.

Liang, Z.; Wang, T.; Zhang, X.; Sun, J.; and Shen, J. 2022. Tree energy loss: Towards sparsely annotated semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16907–16916.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J. 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3159–3167.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. 740–755.

Liu, H.; Peng, C.; Yu, C.; Wang, J.; Liu, X.; Yu, G.; and Jiang, W. 2019. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6172–6181.

Mengya Han, B. Y. Y. L. H. H. B. D.-Y. W. D. T., Yibing Zhan. 2023. Region-adaptive Concept Aggregation for Few-shot Visual Recognition. *Machine Intelligence Research*, 20: 554–568.

Obukhov, A.; Georgoulis, S.; Dai, D.; and Van Gool, L. 2019. Gated CRF loss for weakly supervised semantic image segmentation. *arXiv preprint arXiv:1906.04651*.

Qi Zheng, D. W. D.-C. T., Chao-Yue Wang. 2023. Visual Superordinate Abstraction for Robust Concept Learning. *Machine Intelligence Research*, 20: 79–91.

Shen, R.; Guthier, T.; Mobis, H.; Tang, B.; and Ayed, I. B. 2019. Scribble supervised annotation algorithms of panoptic segmentation for autonomous driving. In *Proc. NeurIPS Workshop Mach. Learn. Auton. Driving*, volume 2.

Shen, Y.; Cao, L.; Chen, Z.; Lian, F.; Zhang, B.; Su, C.; Wu, Y.; Huang, F.; and Ji, R. 2021. Toward joint thing-and-stuff mining for weakly supervised panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16694–16705.

Shi, J.; and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8): 888–905.

Tang, M.; Djelouah, A.; Perazzi, F.; Boykov, Y.; and Schroers, C. 2018a. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1818–1827.

Tang, M.; Perazzi, F.; Djelouah, A.; Ben Ayed, I.; Schroers, C.; and Boykov, Y. 2018b. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 507–522.

Tian, Z.; Shen, C.; Wang, X.; and Chen, H. 2021. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5443–5452.

Wang, B.; Qi, G.; Tang, S.; Zhang, T.; Wei, Y.; Li, L.; and Zhang, Y. 2019. Boundary perception guidance: A scribble-supervised semantic segmentation approach. In *IJCAI International joint conference on artificial intelligence*.

Xiong, Y.; Liao, R.; Zhao, H.; Hu, R.; Bai, M.; Yumer, E.; and Urtasun, R. 2019. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8818–8826.

Zhiqiang Chen, J. L. H. H., Ting-Bing Xu. 2022. Sharing Weights in Shallow Layers via Rotation Group Equivariant Convolutions. *Machine Intelligence Research*, 19: 115–126.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.