Supplementary Material of 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

A Supplementary material of Sec. 3.3

A.1 Further Explanation of $\mathcal{L}^l_{partial}$, $\mathcal{L}^l_{col}$, and $\mathcal{L}^l_{self}$

As mentioned in "**Decoder Layer**" of Sec. 3.3 in our manuscript, the distance branch is supervised by partial cross-entropy loss $\mathcal{L}^l_{partial}$, color-prior loss $\mathcal{L}^l_{col}$, and self-training loss $\mathcal{L}^l_{self}$. Here we offer further explanations of these three losses as follows:

Partial cross-entropy loss: $\mathcal{L}^l_{partial}$ supervises the distance result (probability map) at labeled pixels, we should note that each point label is expanded to a 17×17 square region during the training process (as mentioned in Sec. 4.2), thus the 17×17 square patches around each point label are supervised by $\mathcal{L}^l_{partial}$, which makes the distance branch learn to predict right results at these patches.

Color-prior loss: $\mathcal{L}^l_{col}$ first computes the affinity label $\mathcal{A}_{i,j}$ as follows:

$$\mathcal{A}_{i,j} = \begin{cases} 1 \text{ if } exp\{-\frac{1}{2}\|I^{LAB}[i] - I^{LAB}[j]\|_2\} > \tau \\ 0 \qquad\qquad \text{otherwise} \end{cases}, \tag{A}$$

where I^{LAB} is the LAB color format of input image I, $\|\cdot\|_2$ is the l_2 norm function, the threshold τ is set as 0.3. According to Eq. (A), if two pixels i and j are similar in LAB space, $\mathcal{A}_{i,j} = 1$, otherwise, $\mathcal{A}_{i,j} = 0$. After computing $\mathcal{A}_{i,j}$, we supervise the probability map with Eq. (9) in our manuscript, Eq. (9) is as follows:

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

where $Z^{col} = \sum_{i=1}^{HW} \sum_{j \in \mathcal{N}_i} \mathcal{A}_{i,j}$ is the normalizing factor, $\mathcal{N}_i$ denotes neighboring pixels of pixel i, $M^l[i]$ and $M^l[j]$ are two probability vectors of size $N \times 1$, their l_1 norm is 1, $(M^l[i]^T M^l[j])$ is the inner product of $M^l[i]$ and $M^l[j]$, which depicts the similarity between $M^l[i]$ and $M^l[j]$. When $\mathcal{A}_{i,j} = 1$, $\mathcal{L}^l_{col}$ requires $(M^l[i]^T M^l[j])$ to be as large as possible, which requires $M^l[i]$ and $M^l[j]$ to predict similar results, when $\mathcal{A}_{i,j} = 0$, $(M^l[i]^T M^l[j])$ is not supervised.

In general, $\mathcal{L}^l_{col}$ requires the distance branch to predict similar results for neighboring pixels with similar LAB values, through the constraint between neighboring pixels, the ground truth supervision for labeled pixels can be propagated to unlabeled pixels.

Self-training loss: $\mathcal{L}^l_{self}$ supervises the probability map M^l with the pseudo label $\mathcal{M}^L$ from the last decoder layer in a self-training manner. As mentioned in Sec. 3.3, pseudo labels are improved iteratively, $\mathcal{M}^L$ is more accurate than pseudo labels $\{\mathcal{M}^l\}_{l=1}^{L-1}$ from earlier layers, thus $\mathcal{M}^L$ can improve $\{M^l\}_{l=1}^{L-1}$ through $\{\mathcal{L}^l_{self}\}_{l=1}^{L-1}$. For the last probability map M^L, $\mathcal{L}^L_{self}$ is as follows:

$$\mathcal{L}^L_{self} = \frac{1}{HW}\sum_{i=1}^{HW} CE(M^L[i], \mathcal{M}^L[i]) = -\frac{1}{HW}\sum_{i=1}^{HW} \log M^L_{\mathcal{M}^L[i]}[i], \tag{C}$$

where i is the pixel index, $M^L_{\mathcal{M}^L[i]}[i]$ is the $\mathcal{M}^L[i]$-th channel of $M^L[i]$, $\mathcal{L}^L_{self}$ requires this channel to be larger (more confident). Since most pixel labels in $\mathcal{M}^L$ are right, $\mathcal{L}^L_{self}$ requires the right channels of most pixels to be more confident and reduces the uncertainty of M^L, in this way, M^L is improved by $\mathcal{L}^L_{self}$.

Finally, supervising the distance branch with $\mathcal{M}^L$ can also be seen as a propagation of ground truth supervision at labeled pixels. Specifically, the distance branch can learn to predict right results at labeled pixels under the supervision of $\mathcal{L}^l_{partial}$, then this branch can predict right results at most unlabeled pixels thanks to its generalization ability on unlabeled pixels, thus the pseudo label $\mathcal{M}^L$ based on the prediction of the distance branch offers right labels for most unlabeled pixels. By supervising the distance branch with $\mathcal{M}^L$, the ground truth supervision at labeled pixels is propagated to unlabeled pixels implicitly.

A.2 Detailed Description of the 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**" of Sec. 3.3. Following [5], we predict M^l based on the cross-attention between anchor queries $\widehat{Q}^l$ and multi-scale pixel features F^1, F^2, F^3. As shown in Fig. A we flatten $F^1 \in \mathbb{R}^{D\times H_1\times W_1}, F^2 \in \mathbb{R}^{D_2\times W_2}, F^3 \in \mathbb{R}^{D\times H_3\times W_3}$ into 1D features and concatenate them to get $F \in \mathbb{R}^{L\times D}$, $L = H_1W_1+H_2W_2+H_3W_3$. Then we compute the **cross attention map** $A^l \in [0,1]^{N\times N^h\times L}$ between $\widehat{Q}^l \in \mathbb{R}^{N\times D}$ and F, where N^h is the attention head number. A^l **is the output of the MHCA layer**, it is split into 3 parts along the last dimension ($L \rightarrow [H_1W_1, H_2W_2, H_3W_3]$), these 3 parts are reshaped into **2D maps** $A^l_1 \in [0,1]^{N\times N^h\times H_1\times W_1}$, $A^l_2 \in [0,1]^{N\times N^h\times H_2\times W_2}$, $A^l_3 \in [0,1]^{N\times N^h\times H_3\times W_3}$. A^l_1, A^l_2, A^l_3 are processed by a convolution layer (channel size

N^h is not changed), a Relu layer, and upsampled to size $N \times N^h \times H_1 \times W_1$. Finally, we concatenate 3 upsampled results to get a $N \times 3N^h \times H_1 \times W_1$ feature map and reduce its channel size from $3N^h$ to 1 with a convolution layer to get a $N \times 1 \times H_1 \times W_1$ mask logit, we apply Softmax along the first dimension and get probability map $M^l \in \mathbb{R}^{N \times 1 \times H_1 \times W_1}$, namely $M^l \in \mathbb{R}^{N \times H_1 \times W_1}$. In a word, the output of our MHCA layer is not the result of "V" multiplied by attention maps, but just the attention maps, thus the output size is not equal to the input query size.

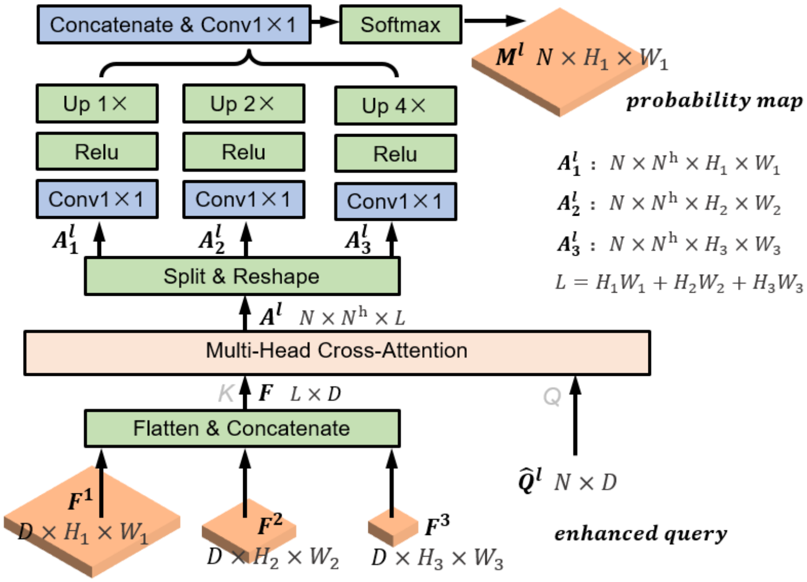
Concatenate & Conv1×1
Softmax
M^l $N \times H_1 \times W_1$
probability map
Up 1×
Up 2×
Up 4×
Relu
Relu
Relu
Conv1×1
Conv1×1
Conv1×1
A_1^l
A_2^l
A_3^l
$A_1^l : N \times N^h \times H_1 \times W_1$
$A_2^l : N \times N^h \times H_2 \times W_2$
$A_3^l : N \times N^h \times H_3 \times W_3$
$L = H_1W_1 + H_2W_2 + H_3W_3$
Split & Reshape
A^l $N \times N^h \times L$
Multi-Head Cross-Attention
K
F $L \times D$
Q
Flatten & Concatenate
$\hat{Q}^l$ $N \times D$
enhanced query
F^1
$D \times H_1 \times W_1$
F^2
$D \times H_2 \times W_2$
F^3
$D \times H_3 \times W_3$

Fig. A: The l-th decoder layer of the distance branch.

B Supplementary material of Sec. 4.2

B.1 More Training Details

During the model training process, we upsample the probability map M^L and pseudo label M^L from size $H_1 \times W_1$ to size $H \times W$, then compute losses $\mathcal{L}^l_{partial}$, $\mathcal{L}^l_{col}$, and $\mathcal{L}^l_{self}$ at $H \times W$ pixels, this helps preserve the details of instances. Besides, the learning rate of the backbone is 1.4e-5 (1/10 of the base learning rate). To reduce the influence of noisy M^L at the first epoch, we multiply a linear warm-up factor to losses that adopt M^L as the training label. On COCO, the batch size is 8 and the model is trained for 12 epochs. , the learning rate is decayed by a factor of 0.1 at the last 4 epochs. On VOC, the batch size is 4 and the model is trained for 20 epochs. the learning rate is decayed by a factor of 0.1 at the last 5 epochs. Each point label is expanded to a 17×17 square region to supply more supervision for the model. The color-prior loss indexes neighboring pixels with a 5×5 kernel and its threshold τ is 0.3.

C Supplementary material of Sec. 4.3

C.1 Comparison of RGB Space and LAB Space in Color Prior Loss

Following previous works [2, 4, 6], we adopt LAB space in the color-prior loss ($\mathcal{L}^{l}_{col}$, Eq. (9) in our manuscript) because LAB space is closer to human perception as stated in [6]. As shown in Tab. A, if we apply color-prior loss in RGB space (the image input format of deep networks), the model performance will drop from 56.6% to 55.9% when trained with the single-point label on VOC [1], demonstrating the effectiveness of LAB space.

Table A: Model performance when applying the color-prior loss in different color spaces.

Color Space	PQ	SQ	RQ
RGB	55.9	82.0	66.9
LAB	56.6	81.4	68.1

C.2 Erosion Operation in Query Aggregating

During the query aggregating process in Sec. 3.3, we remove noisy labels at contour regions of the pseudo label through the erosion operation, this operation is conducted 3 times with a 3×3 kernel by default. Here we analyze the influence of the kernel size and iteration number in Tab. B, the model performs worse when changing the default kernel size 3×3 or iteration number 3 to other values, demonstrating the default erosion setting provides the best balance between noise label removal and full instance coverage.

Table B: Ablations of pseudo label erosion in the query aggregating process.

size	3×3	2×2	4×4	3×3	3×3
#iter	3	3	3	2	4
PQ	56.6	54.0	52.7	55.1	54.3

C.3 Influence of Point Label Numbers

Previous methods PSPS and Point2Mask both show the results of training models with $\mathcal{P}_1$ (one point label per instance) and $\mathcal{P}_{10}$ (ten point labels per instance), thus we mainly show the results of $\mathcal{P}_1$ and $\mathcal{P}_{10}$ for fair comparison in our manuscript. Here we also train models with $\mathcal{P}_2$, $\mathcal{P}_3$, ..., $\mathcal{P}_9$ labels (two, threee, ..., nine point labels per instance) . As shown in Tab. C, the model performance improves very slowly after $\mathcal{P}_7$ label.

Table C: Models' performance when trained with different labels.

label	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	$\mathcal{P}_8$	$\mathcal{P}_9$	$\mathcal{P}_{10}$
PQ	56.6	59.8	62.1	62.3	62.6	62.9	63.6	63.7	63.9	64.0

C.4 Analysis of Distance Map Training Strategy

As mentioned in Sec. 3.3, we get the distance map $X^l \in [0,1]^{N \times H_1 \times W_1}$ from the probability map M^l through Eq. (6) of our manuscript, and supervise M^l to train X^l indirectly. Here we also train our model by directly supervising the distance map X^l. Specifically, for each pixel in X^l, we require the distance to the ground truth instance to be 0, and the distance to other instances to be 1. Given the one-hot format $\mathcal{P}^{01} \in \{0,1\}^{N \times H \times W}$ of point-level instance ID mask $\mathcal{P}$, we can get the distance label $1-\mathcal{P}^{01}$, then we can upsample X^l to size $H \times W$ and supervise X^l with $1 - \mathcal{P}^{01}$ as follows:

$$\mathcal{L}^l_{partial\_dist} = \frac{1}{Z^{partial}} \sum_{\mathcal{P}[i] \neq 255} DIFF(X^l[i], 1 - \mathcal{P}^{01}[i]), \tag{D}$$

where i is the pixel index, 255 denotes unlabeled pixels which are ignored in Eq. (D) , $DIFF(\cdot,\cdot)$ is a function that measures the difference between two vectors, $Z^{partial}$ is the normalizing factor, it is the number of labeled pixels. Similarly, we can also supervise X^l with the one-hot format $\mathcal{M}^{L\text{-}01} \in \{0,1\}^{N \times H \times W}$ of pseudo label $\mathcal{M}^L$ as follows:

$$\mathcal{L}^l_{self\_dist} = \frac{1}{HW} \sum_{i=1}^{HW} DIFF(X^l[i], 1 - \mathcal{M}^{L\text{-}01}), \tag{E}$$

finally, we supervise X^l through the color-prior loss by replacing M^l with X^l:

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

By replacing $\mathcal{L}^l_{partial}$, $\mathcal{L}^l_{col}$, $\mathcal{L}^l_{self}$ with $\mathcal{L}^l_{partial\_dist}$, $\mathcal{L}^l_{col\_dist}$, $\mathcal{L}^l_{self\_dist}$ in $\mathcal{L}^l_{distance}$ in Eq. (11) of our manuscript, we can supervise X^l directly. For the function $DIFF(\cdot,\cdot)$, we tried MSE loss (Mean Square Error loss) and l_1 Norm loss. As shown in Tab. D, training X^l directly with $DIFF$ function (l_1 Norm or MSE) performs worse than training X^l indirectly by supervising M^l in our default setting.

Table D: Model performance when training the distance map with different strategies.

Settings		PQ	SQ	RQ
indirectly	default	56.6	81.4	68.1
directly	l_1 Norm	53.1	81.0	64.2
	MSE	54.7	81.3	65.9

C.5 Influence of Layer Numbers

In "Influence of Layer Numbers" of Sec. 4.3, we analyze the influence of decoder layer numbers of our distance branch. As shown in Tab. 5 of our manuscript, the model performance increases rapidly when the number of layers increases from 1 to 3, then further increases a little when the layer number increases from 3 to 6. Here we also show some estimated pseudo labels of models with 1, 2, 3, and 6 decoder layers in Fig. B. When the distance branch adopts 1 decoder layer, the pseudo label is estimated from the initial anchor query, which is usually biased to local instance parts, thus the estimated pseudo label is incomplete for most instances (*row* L1 of Fig. B). Adopting 2 decoder layers and estimating pseudo labels from anchor queries aggregated according to previous pseudo labels improves the result in some cases (*row* L2 of Fig. B), further adding 1 decoder layer produces satisfactory results in most cases (*row* L3 of Fig. B), and adopting 6 decoder layers produces proper labels in almost all cases (*row* L6 of Fig. B). The above results demonstrate that adopting multiple decoder layers and aggregating new queries with pseudo labels iteratively benefits pseudo label estimation.

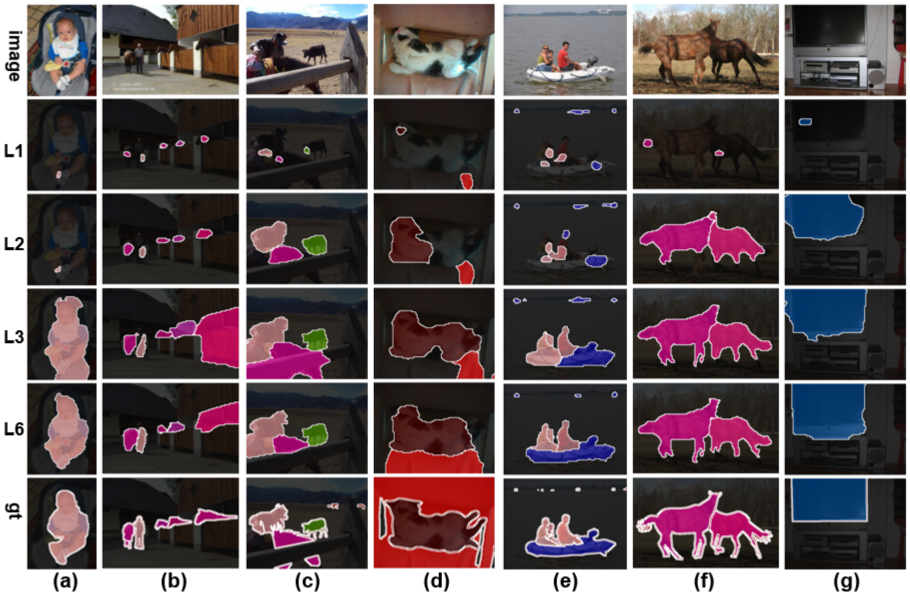

Fig. B: Pseudo labels of models with different decoder layer numbers in the distance branch. "1, 2, 3, 6" denote the model adopts 1, 2, 3, 6 decoder layers, respectively. "gt" denotes the ground truth mask. All models adopt the resnet50 [3] backbone and are trained with $\mathcal{P}_1$ labels.

C.6 Analysis of Decoder Layer Architecture

"layer B" in Fig. C shows the decoder layer of our distance branch (the detailed description of the decoder layer is in Sec. 3.3 of our manuscript), the pseudo label $\mathcal{M}^l$ is estimated based on the cross-attention maps $\{A_i^l\}_{i=1}^3$, the new query $\hat{Q}^{l+1}$ is aggregated using the pseudo label and further enhanced through the query enhancing process, the feed-forward layer is not adopted in our decoder layer. Fig. C also shows another layer architecture “layer A” adopted in [5], “layer A” generates new queries by aggregating features through the cross-attention layer and processing the aggregated result with the feed-forward layer, the new queries are further enhanced through the query enhancing process.

Here we also train our model by adopting “layer A” in the distance branch. As shown in Tab. E, the model performs worse when adopting “layer A”, this is because “layer A” aggregates features based on the cross-attention maps in the cross-attention layer, thus may aggregate noisy features from different instances for one anchor query which aims to represent one instance, our architecture “layer B” aggregates features at the target instance region depicted by the pseudo label, our anchor query contains much less noisy features.

Table E: Model performance when adopting different decoder layer architectures.

Layer arch	PQ	SQ	RQ
layer A	41.4	77.5	51.4
layer B	56.6	81.4	68.1

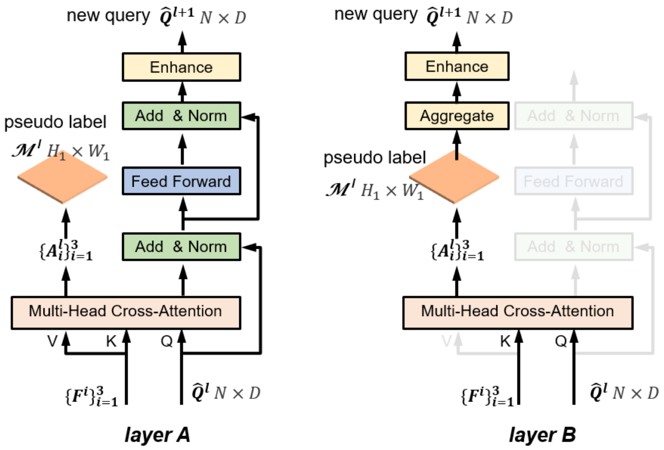

Fig. C: “layer A” predicts the new query $\hat{Q}^{l+1}$ for the next layer by aggregating V features with the cross-attention layer and further processing the aggregated features with the feed-forward layer. In “layer B”, the new query $\hat{Q}^{l+1}$ is aggregated with its predicted pseudo label $\mathcal{M}^l$ directly, the feed-forward layer and input V are discarded.

D Supplementary material of Sec. 4.4

D.1 Pseudo Label Comparison with PSPS

In Fig. 4 of our manuscript, we compare the pseudo label of our model with Point2Mask [4] on VOC [1] *train* set. Here we also compare the pseudo label of our model with PSPS [2] and the result is shown in Fig. D. Our pseudo labels are more accurate than Point2Mask and PSPS in general. Even though there are a few errors in our results when the texture is ambiguous, our model performs much better than PSPS and Point2Mask in these cases (column b, c, d, f of Fig. D). We should note that texture ambiguity is a challenging problem in the segmentation field, even fully-supervised models fail in ambiguous textures. This problem is more challenging when training labels are sparse points. We solve this problem better than PSPS and Point2Mask thanks to our learnable distance.

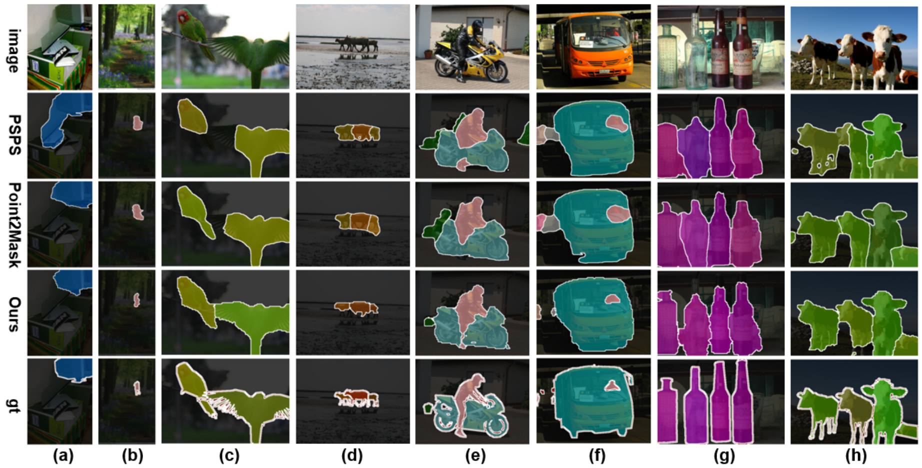

Fig. D: Pseudo labels of PSPS, Point2Mask, and Ours on VOC *train* set, "gt" denotes the ground truth label. Models adopt resnet50 backbone and are trained with $\mathcal{P}_1$ labels.

D.2 Visualization on VOC and COCO

We illustrate some panoptic segmentation results on *val* set of Pascal VOC and COCO in Figs. E and F, our model predicts accurate results in simple scenes and also performs well in complex scenes with multiple small objects.

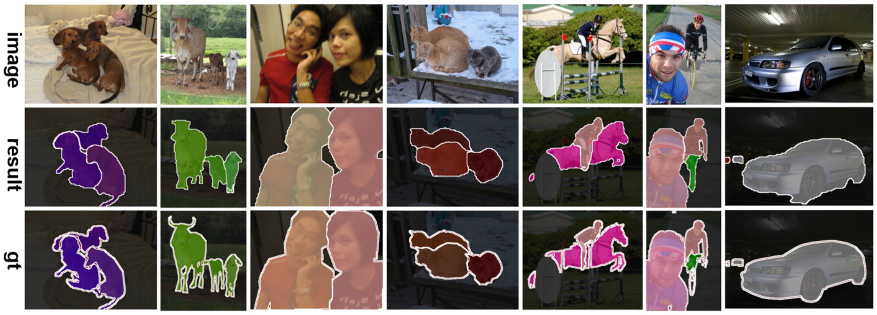

Fig. E: Illustration of panoptic segmentation results on Pascal VOC *val* set, the model adopts Swin-L backbone and is trained with $\mathcal{P}_1$ label (one point label per instance), "gt" denotes the ground truth label.

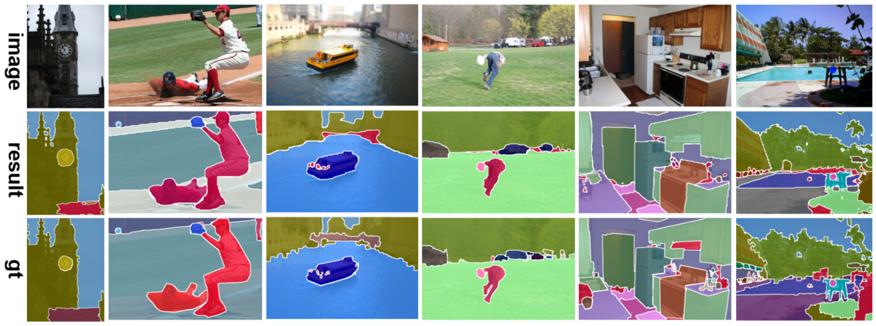

Fig. F: Illustration of panoptic segmentation results on COCO *val* set, the model adopts Swin-L backbone and is trained with $\mathcal{P}_1$ labels (one point label per instance), "gt" denotes the ground truth label.

D.3 Comparison with Related Works in SQ and RQ

In Tab. 8 of our manuscript, we compare our method with related works in PQ, PQ^{th}, and PQ^{st}. Here we also compare our method with PSPS and Point2Mask in PQ, SQ, and RQ, and the results are shown in Tab. F. We can see that our method also significantly outperforms PSPS and Point2Mask in SQ and RQ.

Table F: Comparison with other work. Models are evaluated on *val* set of MS COCO and VOC. $\mathcal{M}$, $\mathcal{B}$, $\mathcal{I}$ 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	SQ	RQ	PQ	SQ	RQ
PSPS [2]	R50	$\mathcal{P}_1$	29.3	73.4	38.8	49.8	78.4	62.0
Point2Mask [4]	R50	$\mathcal{P}_1$	32.4	75.1	41.5	53.8	80.2	65.4
Ours	R50	$\mathcal{P}_1$	34.2	77.4	42.5	56.6	81.4	68.1
Point2Mask [4]	R101	$\mathcal{P}_1$	34.0	75.1	43.5	54.8	79.8	67.0
Ours	R101	$\mathcal{P}_1$	35.2	78.4	43.5	57.8	82.0	69.2
Point2Mask [4]	Swin-L	$\mathcal{P}_1$	37.0	75.8	47.2	61.0	83.5	71.6
Ours	Swin-L	$\mathcal{P}_1$	41.0	79.1	50.3	68.5	84.4	80.0
PSPS [2]	R50	$\mathcal{P}_{10}$	33.1	74.2	42.2	56.6	82.5	67.7
Point2Mask [4]	R50	$\mathcal{P}_{10}$	35.2	75.7	44.9	59.1	82.2	70.6
Ours	R50	$\mathcal{P}_{10}$	41.3	77.9	51.5	64.0	85.0	74.2

References

1. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge **88**, 303–338 (2009)
2. Fan, J., Zhang, Z., Tan, T.: Pointly-supervised panoptic segmentation. In: European Conference on Computer Vision. pp. 319–336. Springer (2022)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2016)

4. 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)
5. 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)
6. 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)