Panoptic-PartFormer: Learning a Unified model for Panoptic Part Segmentation-Supplementary

Xiangtai Li[1] Shilin Xu[1] Yibo Yang[1,3]
Guangliang Cheng[2] Yunhai Tong[1] Dacheng Tao[3]

[1] Key Laboratory of Machine Perception, MOE, School of Artificial Intelligence, Peking University
[2] SenseTime Research
[3] JD Explore Academy
lxtpku@pku.edu.cu, xushilin@stu.pku.edu.cn,chengguangliang@sensetime.com

In this supplementary, we provide the following information in addition to the main paper: more experimental details, more visualization results on four datasets including Cityscapes PPS, Pascal Context PPS, Mapillary [7] and BDD [9].

1 More Experimental Details

Detailed Pretraining Process. *Note that all the baselines [3] for thing and stuff prediction use the COCO [4] pretraining and part of the baselines [10,6] use the Mapillary [7] pretraining.* Thus, for fair comparison, we also pretrain our model on COCO and Mapillary datasets.

For COCO [4] dataset pretraining, all the models are trained following detectron2 settings [8]. We adopt the multiscale training following the previous work [1] by resizing the input images such that the shortest side is at least 480 and at most 800 pixels, while the longest at most 1333. We also apply random crop augmentations during training where the train images are cropped with probability 0.5 to a random rectangular patch which is then resized again to 800 (height), 1333 (width). All the models are trained for 36 epochs.

For Mapillary [7] dataset pretraining, we mainly follow the Panoptic-Deeplab settings [2]. We adopt the multiscale training where the scale ranges from 1.0 to 2.0 of origin image, then we apply a random crop of 1024×2048 patches. The horizontal flip is applied. The pretraining process takes 240 epochs due to limited computation cost. We believe more training interations may lead to better results. The Mapillary pretraining is for fair comparison, since the hybrid models in previous work [3,10,6] all use the Mapillary pretraining for better results.

Detailed pretraining results. For COCO dataset pretraining, our models result in 44.3% PQ (ResNet50), 47.0% PQ (ResNet101) and 52.2 % PQ (Swin-base). For Mapillary pretraning, our model reuslts in 44.3 % PQ (Swin-base [5]). *For fair comparison, we do not pretrain the ResNet50 model.*

Will more interaction number helps? We explore to increase the interaction number of our decoder head. We present more detailed results in Tab. 2. We find there is no performance gain over $I = 3$.

Method	road	swalk	build	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mean PartPQ
Previous hybrid models	98.3	80.4	90.3	37.7	44.0	63.4	58.5	74.5	90.9	41.1	88.8	44.1	45.3	53.3	36.4	49.7	67.9	50.2	51.6	61.4
Panoptic-PartFormer	98.0	78.2	89.5	43.5	44.4	59.3	59.5	74.4	90.5	45.8	90.0	46.0	45.9	50.2	35.1	51.0	75.4	50.5	50.1	61.9

Table 1: **Detailed experiment results on Cityscapes Panoptic validation set.** Our models use Swin-base as backbone.

Setting	PQ	PartPQ	Parameter	GFlops
I=1	58.3	54.2	35.3	163.5
I=2	59.5	55.9	36.2	173.6
I=3(default)	61.6	57.4	37.4	185.8
I=4	61.5	57.3	38.6	195.2

Table 2: Effect of Repeat Number.

Setting	PQ	PartPQ	Parameter	GFlops
K-Net + Part Query	60.8	56.0	37.5M	183.9
K-Net + Part Dense Prediction	60.3	55.8	37.2M	184.9
Ours	**61.6**	**57.4**	37.4M	185.6

Table 3: More comparison experiments on CPP Dataset with K-Net (ResNet50). GFlops is obtained with 1200 $\times$ 800 inputs.

Different with the K-Net. There are mainly two different aspects with K-Net. **Firstly,** we propose a decoupled decoder design rather than shared decoder to refine part queries. Our key insight is the part features should not be *the same as scene features*. This enhances the part segmentation (about 1.0% PartPQ gain, 2.7% Part gain in Tab.4(d)). Moreover, asked by R2, we perform extra experiments using K-Net via adding extra part query in first row of the Tab. 3. We find our design leads to 1.4% PartPQ gain. **Secondly,** rather than directly adding dense part prediction heads (which is done Panoptic-FCN), we propose a joint query learning framework by directly combining part queries into scene queries (thing, stuff). In this way, the part query can also benefit the thing and stuff learning, since part classes pose constraint on low-level details. In the second row of Tab. 3, we show the about 1.6% PartPQ gains over directly K-Net baseline. As shown in Tab.6(c) of the main paper, our joint learning leads to slight improvements on CPP dataset. The above two aspects shows our model is not simple extension of K-Net, and it is *well-designed* for PSS.

Detailed Results on Cityscapes PPS. We present detailed results on PartPQ of each class in the Tab. 1. Our method achieves better results in several classes including train, wall, sky, person, and rider.

2 More Visualization Results

More Visualization results on Cityscapes PPS.

In Fig. 1, we present more examples for Cityscapes Panoptic Part Segmentation. The first four rows show the results on the road driving scene, while the

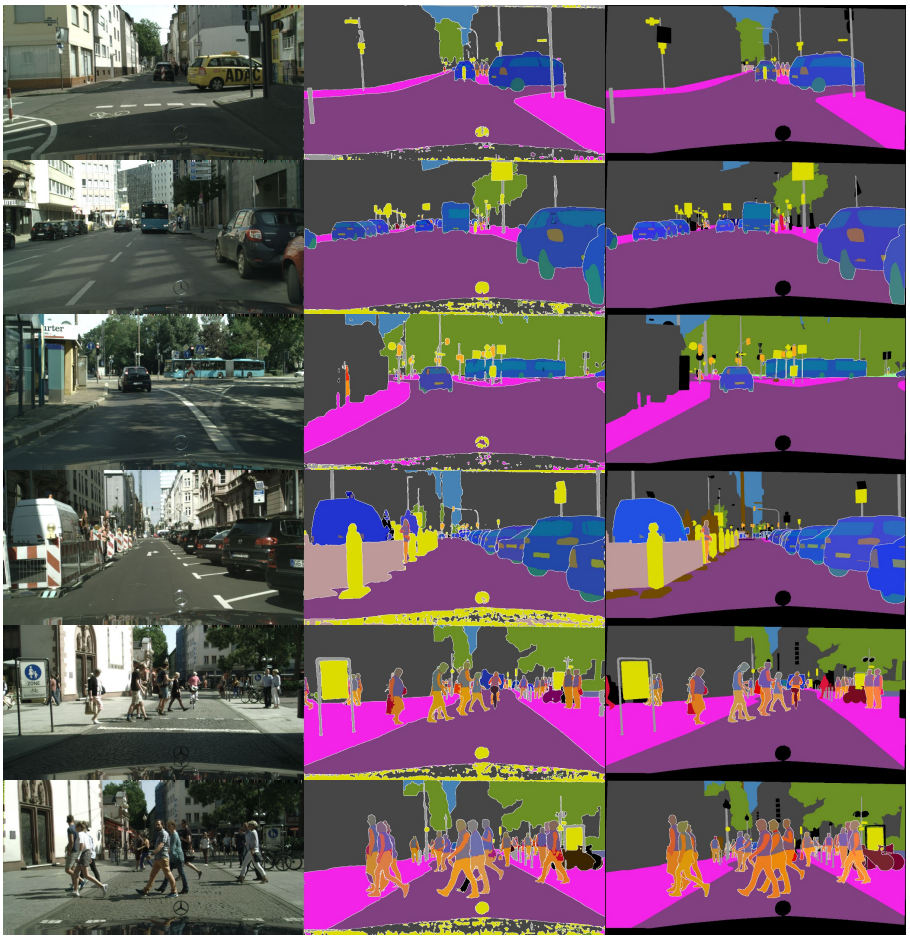

Fig. 1: More visualization results on Cityscaeps Panoptic Part validation set. Best viewed in color and by zooming in. Black regions are ignored during the evaluation. We show the examples of driving car and bus in the first four rows.

last two rows show the crowded human scene. Our Panoptic-Partformer works well on both case. We use the Swin-base model for visualization.

More Visualization results on Pascal Context PPS.

We also visualize several examples on Pascal Context Panoptic Part segmentation dataset in Fig. 2. The left figures show the human scene(including corwded scene). The right figures show several scene that contains the non-human part.

More generalization results on Mapillary and BDD.

In Fig. 3, we give more visual results on generalization on BDD datasets. The second row shows the scene with rainy weather. Our Panoptic-Partformer still

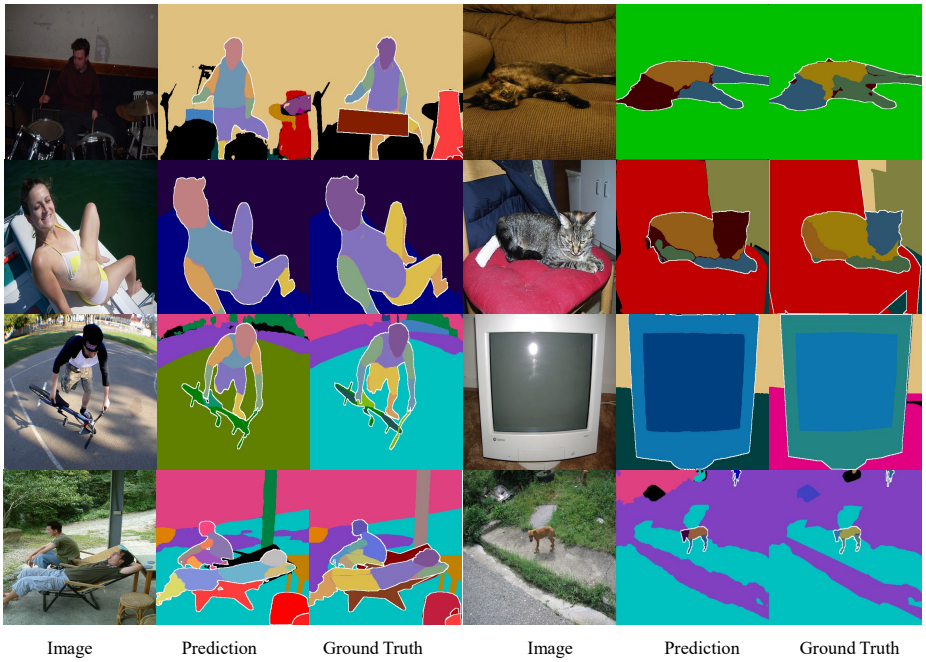

Fig. 2: More visualization results on Pascal Context Panoptic Part validation set. Best viewed in color and by zooming in. *Note that stuff classes have the same color, while thing and part classes are not.*

Fig. 3: More visualization results on BDD dataset. Best viewed in color and by zooming in. We use the color map of Cityscapes for visualization. The first row shows the normal driving cases, while the second row shows the driving cases with different weather (rain).

works well, which proves both robustness and generalization of our method. All the figures are obtained from our Cityscapes models *without* training on BDD dataset. In Fig. 4, we present more results on Mapillary datasets.

Fig. 4: More visualization results on Mapillary dataset. Best viewed in color and by zooming in. We use the color map of Cityscapes for visualization. Both rows show the generalization ability of our approaches.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
2. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR (2020)
3. de Geus, D., Meletis, P., Lu, C., Wen, X., Dubbelman, G.: Part-aware panoptic segmentation. In: CVPR (2021)
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
5. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV (2021)
6. Mohan, R., Valada, A.: Efficientps: Efficient panoptic segmentation. International Journal of Computer Vision **129**(5), 1551–1579 (2021)
7. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017)
8. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
9. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR. pp. 2636–2645 (2020)
10. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. ECCV (2020)