Inferring and Leveraging Parts from Object Shape for Improving Semantic Image Synthesis – Supplementary Materials

Yuxiang Wei[1,2] Zhilong Ji[3] Xiaohe Wu[1(✉)] Jinfeng Bai[3] Lei Zhang[2] Wangmeng Zuo[1]
[1]Harbin Institute of Technology [2] The Hong Kong Polytechnic University [3]Tomorrow Advancing Life
yuxiang.wei.cs@gmail.com {wuxiaohe, wmzuo}@hit.edu.cn cslzhang@comp.polyu.edu.hk

The following materials are provided in this supplementary file:

- Sec. A: more details of the constructed Part Object Dataset (cf. Sec. 4.1 in the main paper).
- Sec. B: more details of the PartNet (cf. Sec. 3.1 in the main paper).
- Sec. C and Sec. D: more ablation studies on PartNet and Synthesis (cf. Sec. 4.4 in the main paper).
- Sec. E: more qualitative comparison results (cf. Sec. 4.2 in the main paper).
- Sec. F: more quantitative comparison results (cf. Sec. 4.3 in the main paper).

A. Object Part Dataset

A.1. Dataset

We build the object part dataset based on the Cityscapes PPS and Pascal VOC PPS datasets proposed by Geus *et al*. [3]. Specifically, each object part is cropped, resized to 64×64 based on its bounding box, and formed as paired (object shape, object part map). For categories in Pascal VOC PPS, some parts are merged as one for simplicity. For example, we have combined the parts of quadrupeds into four parts (*i.e*., head, torso, leg, and tail), while combining the parts of cars into five parts (*i.e*., window, wheel, light, license, and chassis). The final annotated parts of each category are reported in Table. B. There is a total of 21 categories in the constructed dataset. For images in Cityscapes PPS, we form the training/testing set based on its official training/validation partition. For images in Pascal VOC PPS, we merge all images and apportion them into training and testing sets, with an 80-20 split. Following [7], the total categories are split into basis categories (20 categories, *e.g*., human, car, bus, and sheep, *etc*.) and novel category (1 category, *i.e*., cat). The training set of basis categories is used to train the PartNet, and the testing set of both basis and novel categories are used to test it. Besides, for those novel categories in semantic image synthesis datasets, we have annotated k part maps manually as supports to perform part prediction, including the washer, van, zebra, cat, and stop sign. The total basis/novel/SIS categories are listed in Table A. Fig. A also illustrates the examples of annotated SIS support part maps.

Table A. Category split for object part dataset. We use the training set of basis categories to train the PartNet, and test it on the testing sets of both basis and novel (validation) categories. Novel classes (SIS Testing) denotes the novel classes used for semantic image synthesis.

Basis Classes	Aeroplane, Bicycle, Bird, Boat, Bottle, Bus, Car, Chair, Cow, Table, Dog, Horse, MotorBike, Human, PottePlant, Sheep, Sofa, Television, Train, Truck
Novel Classes (Validation)	Cat
Novel Classes (SIS Testing)	Washer, Van, Stop sign, Zebra, Cat

Cat Zebra Stop Sign Washer Van

Figure A. Examples of annotated support part maps for semantic image synthesis.

A.2. Selection of Support Part Maps

To obtain the support object part maps for each category, we use k-means to cluster the training object shapes into k clusters based on the shape similarity metric [9]. To measure the similarity between two object shapes (O_i and O_j), we adopt the geometric score [9] to measure shape consistency,

$$\sigma(O_i, O_j) = \frac{\|O_i - O_j\|_2^2}{max\left(\|O_i\|_1, \|O_j\|_1\right))}. \quad (1)$$

Lower $\sigma(O_i, O_j)$ indicates more similarity between two object shapes. After clustering, part maps with the corre-

Table B. Part Annotation Labels

aeroplane	body, stern, wing, wheel
bicycle	wheel, saddle, handlebar, other
bird	head, torso, leg, tail
boat	boat
bottle	cap, body
bus	window, wheel, light, license, chassis
car	window, wheel, light, license, chassis
cat	head, torso, leg, tail
chair	chair
cow	head, torso, leg, tail
table	table
dog	head, torso, leg, tail
horse	head, torso, leg, tail
motorbike	wheel, handlebar, saddle, headlight
human	head, torso, leg, arm
pottedplant	pot, plant
sheep	head, torso, leg, tail
train	headlight, torso
sofa	sofa
tvmonitor	screen, frame
truck	window, wheel, light, license, chassis
washer	door glass, door, machine body
van	window, wheel, light, license, chassis
stop sign	word, octagon
zebra	head, torso, leg, tail

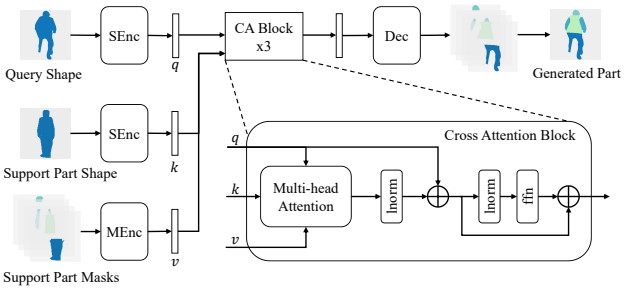

Figure B. Illustration of our PartNet. The support part map is first decomposed into the support part shape and the support part masks as inputs. Cross attention block is adopted to aggregate the part information from the support features.

sponding object shape closest to the cluster center are selected as support part maps.

B. Details of Part Prediction Network

B.1. Network Architecture

As shown in Fig. B, our PartNet takes the query shape $O_q \in \mathbb{R}^{64\times64\times3}$, support part shape $O^S_{y_q} \in \mathbb{R}^{64\times64\times3}$, and support part masks $O^M_{y_q} \in \mathbb{R}^{64\times64\times1}$ as inputs to predict the part map $P_q \in \mathbb{R}^{64\times64\times1}$. In particular, it consists of the shape and mask encoders, cross attention blocks, and the decoder, which will be introduced in the following.

Shape Encoder. The shape encoder is composed of 5 Conv-BN-ReLU layers, where Conv denotes the convolutional layer and BN is the batch normalization layer. For each convolutional layer, the stride is set to 2 to downsample the features. As mentioned in the main paper, the encoders adopt the multi-scale mechanism to perceive the pixels' relative position of the whole object shape. Specifically, the outputs of the last two layers are upsampled and concatenated with the output of the third layer as the final output. Experiments have demonstrated its effectiveness (see Sec. C).

Mask Encoder. The mask encoder adopts the same architecture as the shape encoder except for the different input channels.

Decoder. The decoder consists of 3 DeConv-BN-ReLU layers, where DeConv denotes the transpose convolutional layer. For each transpose convolutional layer, the stride is set to 2 to upsample the features.

B.2. Learning Objective

To facilitate the PartNet learning, we adopt two losses during training. Firstly, a BCE loss is introduced to encourage the predicted part map to be similar to the ground-truth part map,

$$\mathcal{L}_{pre} = \text{BCE}(\text{PartNet}(O_q, S_{y_q}), P^{gt}_q), \tag{2}$$

where P^{gt}_q denotes the ground-truth part map. Besides, when the support shape is the same as the query shape, the predict parts should also be the same as the support parts,

$$\mathcal{L}_{rec} = \text{BCE}(\text{PartNet}(O_q, P^{gt}_q), P^{gt}_q). \tag{3}$$

The final learning objective for PartNet is,

$$\mathcal{L}_{part} = \mathcal{L}_{pre} + \mathcal{L}_{rec}. \tag{4}$$

B.3. Experimental Details

We train our PartNet on one Tesla V100 GPU and adopt Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.999$ where the learning rate is set to 0.0001. The number of support part maps is set to $k = 3$. Our PartNet is pre-trained for 30 epochs and fixed during the synthesis training. Pixel ACcuracy (AC) is adopted as the metric to evaluate the PartNet.

C. More Ablation Studies on PartNet

Effectiveness of Multi-Scale Encoder. We first conduct the ablation study on PartNet to verify the effectiveness of the introduced multi-scale encoder. For comparison, we also train the PartNet without the multi-scale mechanism that the encoders only consist of 3 layers. As shown in Fig. C, without the multi-scale encoder to perceive the

Table C. Ablation studies on the multi-scale encoder. With or w/o MS denotes the PartNet with or without the multi-scale encoder. Basis AC and Novel AC denote the testing accuracy on the basis and novel categories of the object part dataset, respectively.

Object Loss	Basis AC (↑)	Novel AC (↑)
w/o MS	94.05	83.21
with MS	**94.38**	**85.00**

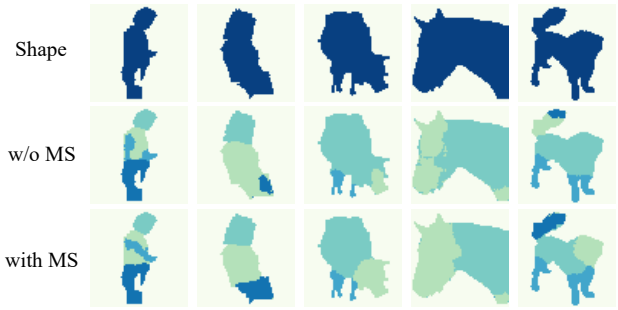

Figure C. Visual comparisons on the effect of the multi-scale encoder. With or w/o MS denotes the PartNet with or without the multi-scale encoder.

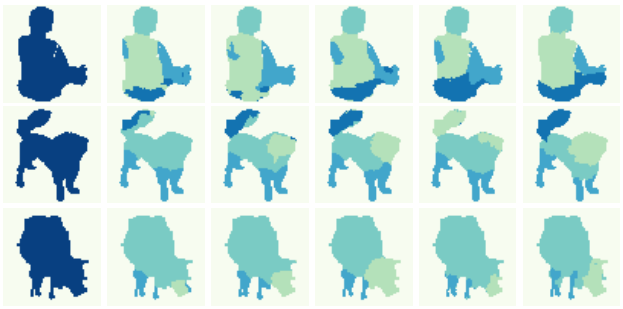

Figure D. Visual comparisons on the effect of different numbers of cross attention blocks.

Table D. Ablation study on the number of cross attention blocks. n denotes the number of blocks.

Methods	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Basis AC (↑)	94.10	94.25	**94.38**	94.34
Novel AC (↑)	82.28	83.76	**85.00**	83.41

whole object shape, the predicted part maps are usually incomplete and discontinuous. In contrast, with the multi-scale encoder, our PartNet predicts more plausible and realistic part maps, and also achieves better performance on both basis and novel categories (see Table. C).

Effectiveness of the number of CA blocks. Furthermore, the effect of the number of cross attention (CA) blocks is also analyzed. We train the PartNet with the different number of CA blocks, and the results are listed in Fig. D and Table D. From Table D, more cross attention blocks bring more prediction capabilities to the PartNet, resulting in better prediction accuracy on both basis and novel categories. However, when the number $n > 3$, the PartNet tends to be overfitting, and adding more cross attention blocks will not bring more performance gain. Thus, we choose PartNet with three cross attention blocks as our final part prediction model.

Table E. Ablation study on the object-level CLIP style loss, and we further compare it with an object-level VGG loss.

Object Loss	FID(↓)	mIOU(↑)	AC(↑)
VGG Loss	41.6	68.6	81.7
CLIP Loss	**41.3**	**70.6**	**82.2**

Table F. Ablation studies of losses and part map on Cityscapes.

Part & PSM	$\mathcal{L}^g_{G/D}$	$\mathcal{L}_{style}$	FID(↓)	mIOU(↑)	AC(↑)	obj FID (↓)
			47.7	66.9	81.5	44.1
✓			44.2	69.0	81.8	36.8
	✓		43.6	66.7	81.9	39.2
		✓	45.2	69.1	81.2	38.9
	✓	✓	42.8	70.5	82.1	37.5
✓		✓	43.1	**70.6**	82.0	35.5
✓	✓	✓	**41.3**	**70.6**	**82.2**	**30.4**

D. More Ablation Studies on Synthesis

CLIP Style Loss vs. VGG Loss. To demonstrate the effectiveness of the object-level CLIP style loss [11], we further compare it with an object-level VGG loss [10]. Specifically, for CLIP style loss, we adopt the pre-trained CLIP image encoder (VIT-32) [5] as the feature extractor, and the tokens of the eighth layer are used to calculate the loss. Each object of the generated images is cropped and resized to 224×224 as input. For VGG loss, we adopt the pre-trained VGG19 [8] as the feature extractor, and the intermediate features are used to calculate the loss. Each object of the generated images is cropped and resized to 128×128 as input. The results are listed in Fig. E and Table E. As shown in Fig. E, benefited from the large-scale pre-training of CLIP, our iPOSE trained with CLIP style loss achieves better visual quality. Besides, it has the ability to refine the part map to generate images with more realistic parts. From Table E, our iPOSE with CLIP style loss also performs better than VGG loss, demonstrating its effectiveness.

Predicted Part Map vs. SPD. We have conducted the experiment by replacing our part map with SPD feature [4] on

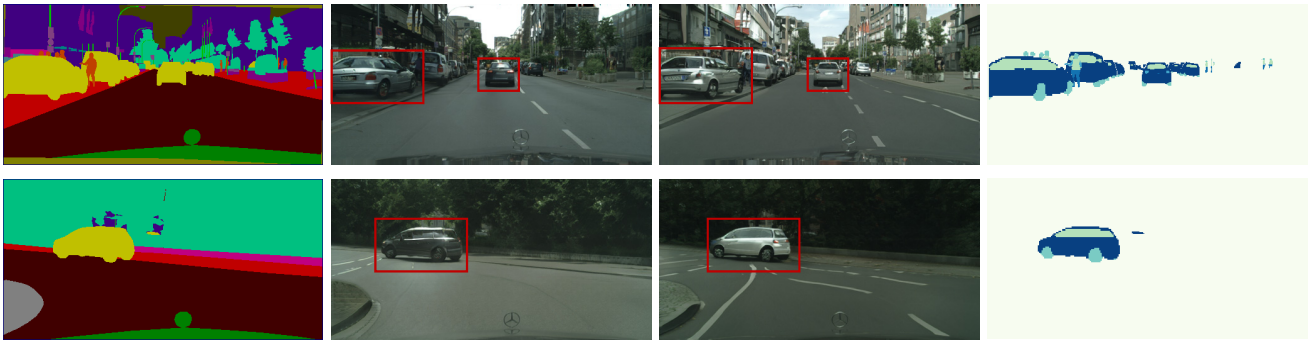

Figure E. Visual comparisons between CLIP style loss and VGG loss.

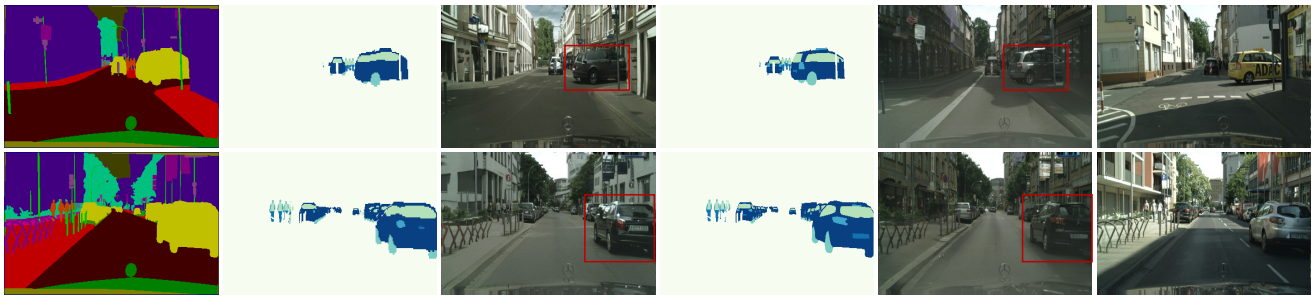

Figure F. Visual comparisons between the images generated with predicted part maps and the images generated with ground-truth part maps.

Table G. Comparison of our part map and SPD feature [4].

Method	FID(↓)	mIOU(↑)	AC(↑)	Method	FID(↓)	mIOU(↑)	AC(↑)
Ours	**41.2**	**70.6**	**82.2**	Ours w/SPD	43.1	70.3	82.0

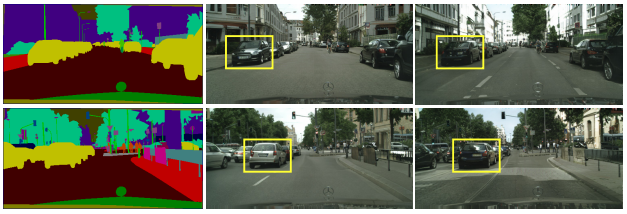

Figure G. Visual comparisons between the images generated with predicted part maps and the images generated with SPD [4].

Cityscapes, while keep same network architecture. From Fig. G(a) and Table G, our iPOSE generates more realistic parts, and also performs favorably against SPD, especially on FID.

Predicted Part Map vs. GT Part Map. We also compare the predicted part maps with the ground-truth part maps.

Table H. Quantitative comparisons between the results generated by the predicted part maps and the results generated by the ground-truth part maps.

Part Map	FID(↓)	mIOU(↑)	AC(↑)
Ours	41.3	70.6	82.2
GT	**40.8**	**70.9**	**82.3**

Since there are part annotations on Cityscapes [3], we additionally train a model with ground-truth part maps as inputs for comparison. The results are listed in Fig. F and Table H. As shown in Fig. F, with the ground-truth part map as guidance, the model generates images with accuracy parts. While our iPOSE with the predicted part map can also generate images with realistic parts, even for those objects with extreme poses (the first row in Fig. F). From Table H, our iPOSE achieves comparable performance compared to the ground-truth part map as input.

More ablations on losses and Part&PSM. We have conducted the experiments by adding Part&PSM, $\mathcal{L}_{style}$, and $\mathcal{L}^{g}_{G/D}$ to baseline respectively, and the results are reported in Table F (first 4 rows). Among them, our Part&PSM con-

Table I. Object-level user study on different datasets. The numbers indicate the percentage (%) of volunteers who favor the results of our method over those of the competing methods.

Dataset	Ours vs. SPADE	Ours vs. CC-FPSE	Ours vs. OASIS	Ours vs. SAFM
Cityscapes [2]	85.3	81.1	80.1	79.2
ADE20K [12]	81.0	74.5	71.7	60.7
COCO-Stuff [1]	75.2	63.1	66.5	72.0

tributes most to the object synthesis and bring a significant performance improvement, especially on object-level FID. $\mathcal{L}^g_{G/D}$ brings more improvement on FID, because it improves not only objects, but also the background. We have further conducted the experiments by adding Part&PSM, $\mathcal{L}_{style}$, and $\mathcal{L}^g_{G/D}$ sequentially. From Table F (rows 1, 2, 6, and 7), our Part&PSM enables to generate photo-realistic object parts and obtains a significant improvement on FID, mIOU, and obj FID. $\mathcal{L}_{style}$ and $\mathcal{L}^g_{G/D}$ can further boost the performance.

E. More Qualitative Results

Fig. H, Fig. I and Fig. J illustrate the qualitative comparisons between our iPOSE and state-of-the-art methods [4, 6]. As shown in the figures, our iPOSE generates images with more realistic parts, further demonstrating its superiority. Moreover, as shown in Fig. K, by sampling different noises for each object, our method can synthesize diverse object results.

F. More Quantitative Results

We have also conducted the object-level user study to evaluate the effects of our iPOSE on object synthesis. Specifically, we cropped and resized each object to 128×128 to perform the object-level human evaluation. From Table I, users tend to favor our results on all the datasets. Besides, compared to the global-level evaluation (Table 2 in main paper), our iPOSE obtains a better preference in object-level.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. 5, 8

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 5, 6

[3] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5494, 2021. 1, 4

[4] Zhengyao Lv, Xiaoming Li, Zhenxing Niu, Bing Cao, and Wangmeng Zuo. Semantic-shape adaptive feature modulation for semantic image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11214–11223, 2022. 3, 4, 5

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[6] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning Representations*, 2021. 5

[7] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 1

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[9] Hao Wang, Qilong Wang, Hongzhi Zhang, Jian Yang, and Wangmeng Zuo. Constrained online cut-paste for object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 1

[10] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 3

[11] Yabo Zhang, Mingshuai Yao, Yuxiang Wei, Zhilong Ji, Jinfeng Bai, and Wangmeng Zuo. Towards diverse and faithful one-shot adaption of generative adversarial networks. *arXiv preprint arXiv:2207.08736*, 2022. 3

[12] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 5, 7

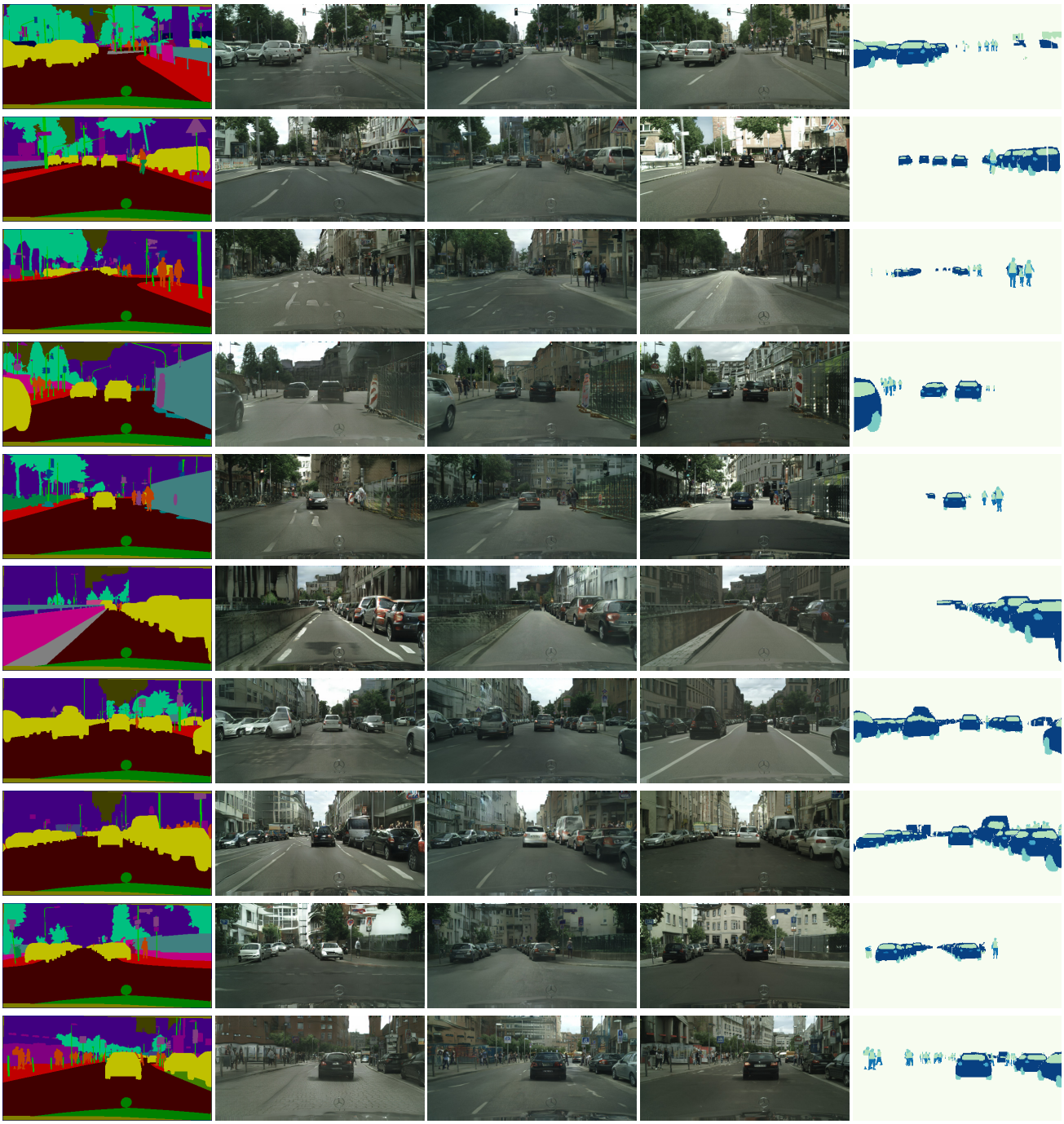

Figure H. More qualitative comparison results on Cityscapes [2].

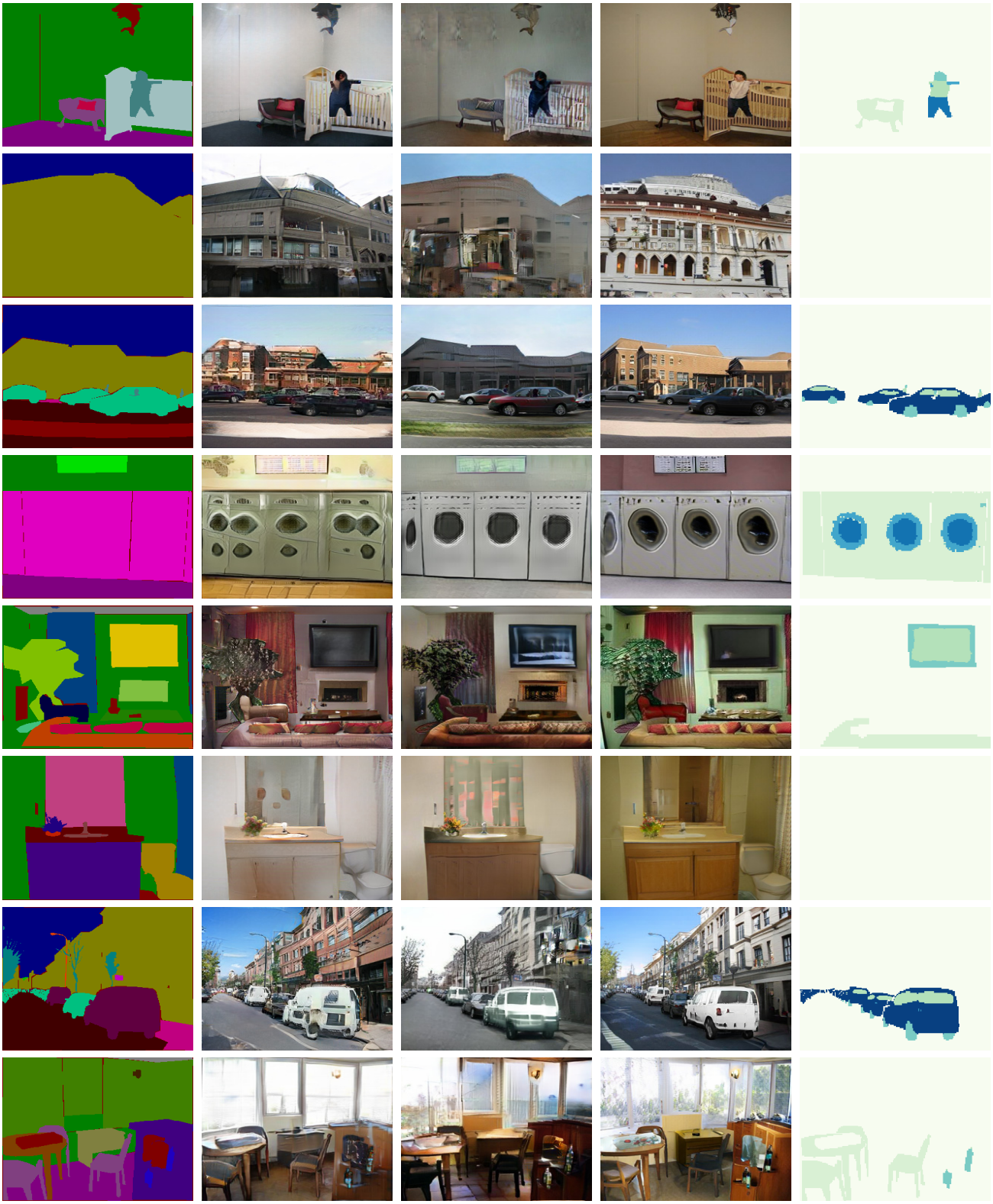

Figure I. More qualitative comparison results on ADE20K [12].

Semantic Map | OASIS [6] | SAFM [4] | Ours | Part Map

Figure J. More qualitative comparison results on COCO [1].

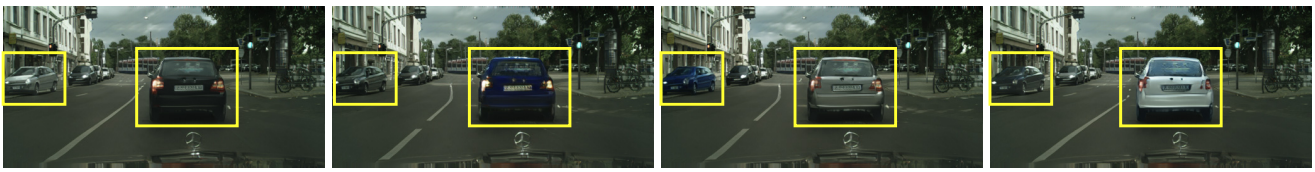

Figure K. Object-level diversity. By sampling different noises for each object, our method can synthesize diverse object results.