A Appendix

In this section, we show more evaluation results on the validation set to demonstrate the effectiveness of the proposed NRD. Visualization results are also included. The code and pre-trained model weights can be found from this link: `https://tinyurl.com/SegNRDNet`.

A.1 More Evaluation Results on the Validation Set

Evaluation on Cityscapes. Table 1 shows the comparison of mIOU performance for different methods on Cityscapes `val` split. The GFlops and the mIOU performance are all measured in the same structure implemented in mmseg [1]. From the comparison we can see that NRD has a very efficient computational cost and performance trade-off. NRD usually has the best performance among methods that are of similar computational cost. Even if compared with the methods that use dilated backbones which have 5 times the computational cost, NRD is still competitive. All the results are plotted in Figure 2 (a) of the main paper.

Table 1: Accuracy and computational costs of different networks on Cityscapes `val` split. The model mIoU is measured at single scale inference. The GFLops is measured at single scale inference with a crop size of 1024×2048. The results are from [1] which provides re-implementation of all the models in the table.

method	backbone	GFlops	mIOU
PSANet	Dilated-ResNet-50	1597.15	77.24
PSANet	Dilated-ResNet-101	2218.62	79.31
PSPNet	Dilated-ResNet-18	434.11	74.87
PSPNet	Dilated-ResNet-50	1427.47	78.55
PSPNet	Dilated-ResNet-101	2048.95	79.76
DeepLabv3+	Dilated-ResNet-18	433.9	76.89
DeepLabv3+	Dilated-ResNet-50	1410.86	80.09
DeepLabv3+	Dilated-ResNet-101	2030.3	**80.97**
OCRNet	HRNetV2p-W18-Small	353.47	77.16
OCRNet	HRNetV2p-W18	424.29	78.57
OCRNet	HRNetV2p-W48	1296.77	80.7
NRD (**Ours**)	ResNet-18	**95.7**	77.5
NRD (**Ours**)	ResNet-50	234.6	79.8
NRD (**Ours**)	ResNet-101	390.0	80.7

A.2 More Visualization Results

Comparison with bilinear upsampling. Fig. 1 shows the results comparison between the bilinear decoder and NRD decoder. Note that those two structures do not evolve low-level features or guidance maps. The results are generated purely from feature maps that are $1/32$ of the input scale. Thus, the results can represent the effectiveness comparison between bilinear method and NRD. From the illustration we can see that it is inevitable for the decoder to loss some details during a 32 times upsampling, however, NRD clearly preserves more detail information than bilinear interpolation method even though the computational cost for those two are similar.

Comparison with the DeeplabV3+ decoder. Fig. 2 shows more comparison between DeeplabV3+ [2] decoder and NRD. Note that for the computational cost of decoder part only, the GFlops of those two are 76.4 (DeeplabV3+) vs 20.4 (NRD). From the figure, we can see that in various scenes, NRD shows superior segmentation result than DeeplabV3+ decoder.

Competitive segmentation results on ADE20K and PASCAL-Context. Fig. 3 is the segmentation results on ADE20K produced by NRD using ResNeXt101 backbone with multi-scale inference. We can see that in various scenes, including the bedroom, the toilet, and some outdoor scenes, NRD can generate satisfying segmentation results. It can also performs well on details, like the human legs and the poles. Fig. 4 is the segmentation results on PASCAL-Context produced by NRD using Resnet101 backbone with multi-scale inference (60 classes). From the illustration we can see NRD successfully captures the boundaries of objects of various shapes.

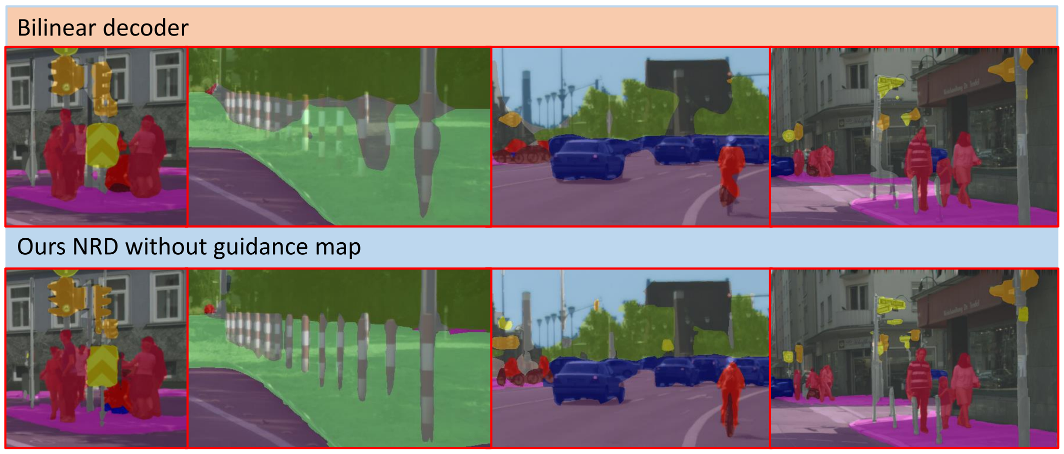

Figure 1: Illustration of the difference between the bilinear decoder and the NRD decoder without guidance map on the Cityscapes dataset. The single scale GFlops of those methods are 215.5 (bilinear decoder) vs 216.8 (NRD decoder), which are almost the same. From the comparison we can see that there is a significant improvement at the boundary region, indicating that our proposed NRD has a very strong representation ability.

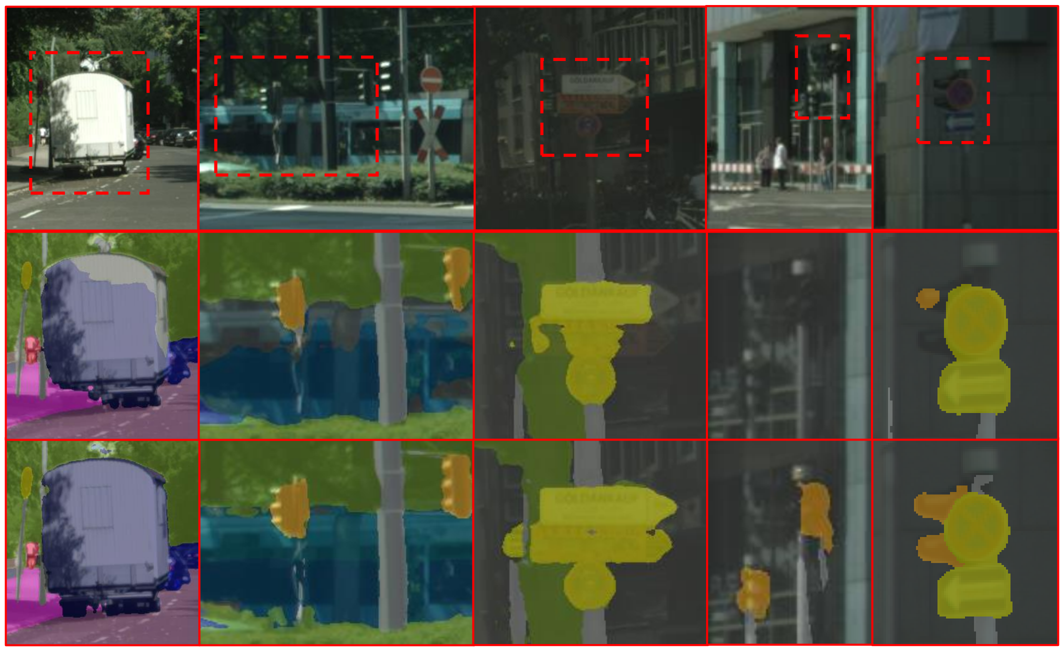

Figure 2: Visualization results on Cityscapes. **First row**: The input images. **Second row**: Zoomed-out segmentation results of the DeeplabV3+ decoder. **Third row**: Zoomed-out segmentation results of NRD decoder. The single scale GFlops of these two methods are 293.6 (DeeplabV3+) vs 234.6 (NRD). NRD requires less computational cost, and the performance at various regions are better.

Detailed illustration of the NRD module. Fig. 5 shows how the image is processed in NRD using real data examples. For the NRD structure, the guidance maps that generated from the low level features as well as the coordinate maps are concatenated together. These feature maps are served as the input to the NRD structure. We attribute each patch of the feature maps with a representational network $g_{\boldsymbol{\theta}}(\cdot)$ whose parameters are dynamically generated by the controller. In Fig. 5 each block surrounded by white lines represents a patch that is processed by a particular $g_{\boldsymbol{\theta}}(\cdot)$. The result is then directly used as output of the decoder without the use of more convolutions to 'refine' the results.

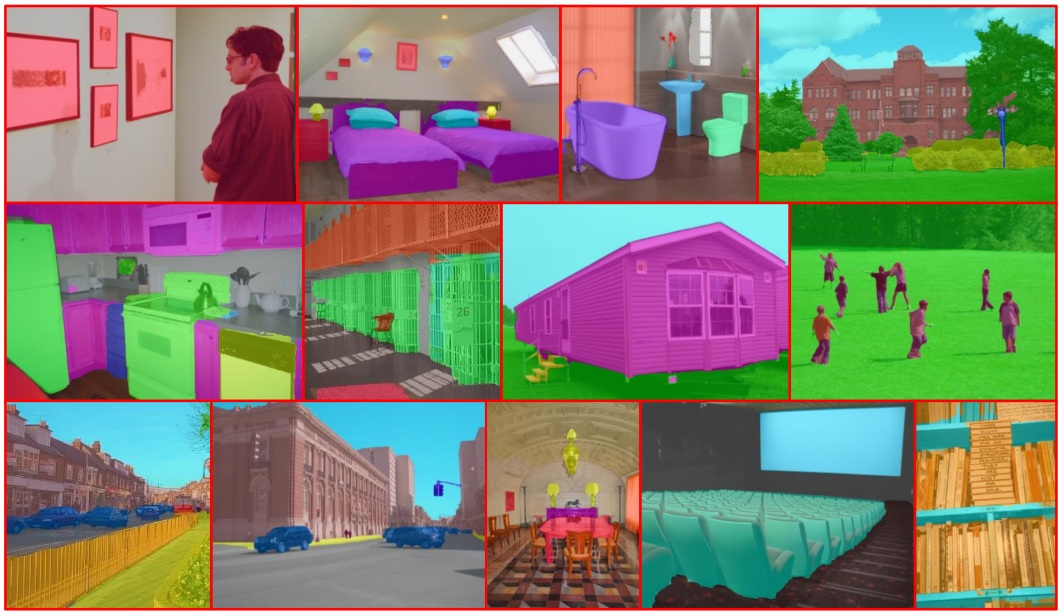
Figure 3: Competitive segmentation results on the ADE20K dataset. Our method performs well on various scenes, and can captures the detailed boundaries information.

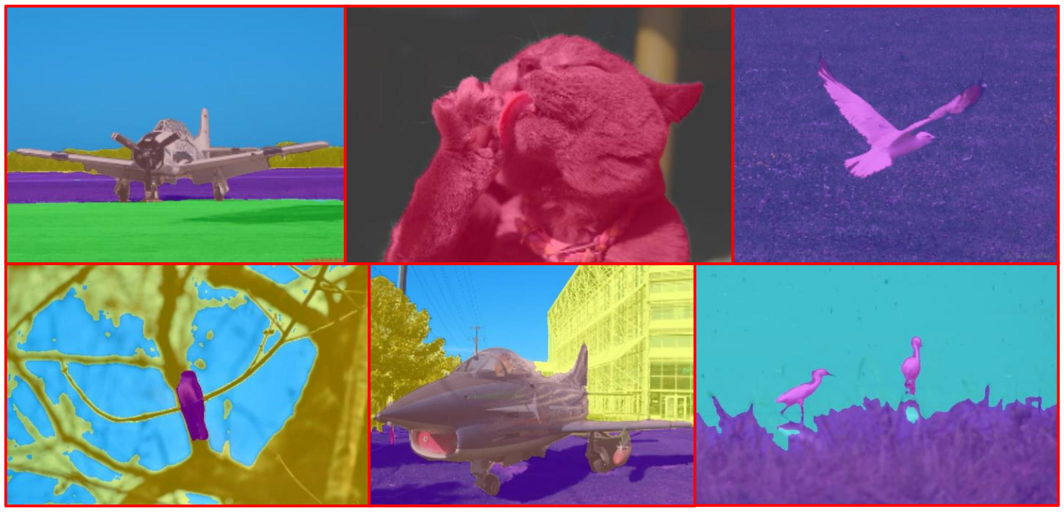
Figure 4: Competitive segmentation results on the PASCAL-Context dataset. The proposed method performs well on various shapes of objects.

References

[1] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. Eur. Conf. Comp. Vis.*, 2018.

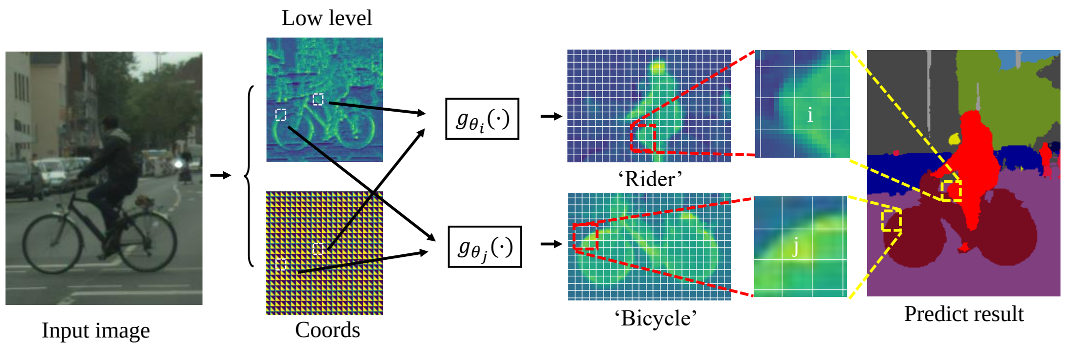

Figure 5: Detailed Illustration of the NRD module. Guidance maps from low-level feature maps as well as coordinate maps are concatenated together and pass through the representational networks $g_{\boldsymbol{\theta}}(\cdot)$. The 'Rider' and 'Bicycle' maps are the illustration of the result feature maps produced by the neural representation. We can see for different patches 'i' and 'j', there are different representational networks $g_{\boldsymbol{\theta}}(\cdot)$ to generate them. From the enlarged map we can see even though the results are generated by a patch-wise manner, the transition of adjacent patches are still smooth.