EDAPS: Enhanced Domain-Adaptive Panoptic Segmentation
Supplementary Material

Suman Saha* ETH Zurich
Lukas Hoyer* ETH Zurich
Anton Obukhov ETH Zurich
Dengxin Dai Huawei Zurich Research Center
Luc Van Gool ETH Zurich, KU Leuven

1. Overview

This supplementary material provides a more detailed analysis of the experiments presented in the paper. In particular, Sec. 2 provides further implementation details, Sec. 3 presents a detailed class-wise performance comparison on additional UDA benchmarks, Sec. 4 highlights the benefits of self-training over adversarial training for UDA panoptic segmentation, Sec. 5 presents an ablation study showing the significance of different instance losses on the adaptation process, Sec. 6 analyzes additional qualitative example predictions, and Sec. 7 offers a visual comparison of the predictions made by EDAPS and M-Dec-BU.

2. Further Implementation Details

EDAPS is implemented in PyTorch [8] based on the DAFormer framework [6]. The source code is available at https://github.com/susaha/edaps to ensure easy reproducibility and promote research in domain-adaptive panoptic segmentation. We follow CVRN [7] and consider 11 stuff-classes and 8 thing-classes. The stuff classes are road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, and sky; the thing classes are person, rider, car, truck, bus, train, motorcycle, and bicycle.

We use a threshold of 0.95 to select the top-k binary masks predicted by the EDAPS instance head. We use these top-k predicted masks to generate the class-agnostic instance segmentation maps, which are then fused with the predicted semantic segmentation maps by a majority-voting rule.

For the Foggy Cityscapes dataset [9], we use the attenuation coefficient $\beta = 0.02$. It specifies the meteorological optical range (MOR) or the visibility, and it is measured in inverse meters. $\beta = 0.02$ corresponds to a MOR of 150m and represents a considerable domain gap to clear weather scenes (see Fig. 8).

M-Dec-BU (Baseline). Since the bottom-up instance decoder (used in the M-Dec-BU) does not directly predict instance masks, a post-processing step is required to generate the class-agnostic instance segmentation maps from the predicted center and offset heatmaps. The post-processing step includes selecting the top-k instance centers and grouping pixels based on these selected centers. We pick the top-k predicted centers by first applying a hard thresholding to filter out the low-confident center predictions following a 2D max pooling on the predicted center heatmap. In all our experiments, we set the threshold to 0.1, max-pooling kernel size to 7×7, and $k = 200$ as in [3].

Once the top-k instance centers are selected, we assign each pixel an instance id based on the predicted offset heatmap. More specifically, the instance id for a pixel is the index of the closest instance center after moving the pixel location by the offset. We filter out the stuff pixels based on the predicted semantic segmentation. Once the instance ids are computed, we generate the class-agnostic instance segmentation maps and fuse them with the predicted semantic segmentation maps by a majority-voting rule [3].

3. Comparison on Additional Benchmarks

In this section, we report UDA performance on additional clear-to-foggy and real-to-real UDA benchmarks. Tab. 1 and Tab. 2 present comparisons with state-of-the-art methods on Cityscapes $\rightarrow$ Foggy Cityscapes and Cityscapes $\rightarrow$ Mapillary Vistas benchmarks. We report a detailed class-wise PQ comparison to gain a better insight into the performance analysis. EDAPS shows significant performance gains for most of the *thing* and *stuff* classes. Most importantly, EDAPS significantly improves the mean recognition quality (mRQ) on both clear-to-foggy and real-to-real benchmarks with a respective percentage gain of 42% (Tab. 1) and 25% (Tab. 2).

4. Adversarial- vs. Self-Training

We chose self-training over adversarial training because it is the predominant SOTA approach in UDA semantic segmentation. Further, adversarial training is rather unstable,

*These authors contributed equally to this work.

Table 1: Comparison with state-of-the-art methods on Cityscapes → Foggy Cityscapes benchmark for UDA Panoptic Segmentation. For clarity, per class PQs are reported. The results of EDAPS are averaged over 3 random seeds.

UDA Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	m.bike	bike	mSQ	mRQ	mPQ
UniDAPS-Baseline [1]	92.5	48.9	60.6	6.0	10.7	5.3	9.9	23.6	49.7	55.6	22.3	15.4	38.5	23.7	1.6	2.8	70.0	38.6	29.2
DAF [2]	**94.0**	54.5	57.7	6.7	10.0	7.0	6.6	25.5	44.6	59.1	26.7	16.7	42.2	36.6	4.5	16.9	70.6	41.7	31.8
FDA [13]	93.8	53.1	62.2	8.2	13.4	7.3	7.6	28.9	50.8	49.7	25.0	22.6	42.9	36.3	10.3	15.2	71.4	43.5	33.0
AdvEnt [11]	93.8	52.7	56.3	5.7	13.5	10.0	10.9	27.7	40.7	57.9	27.8	29.4	44.7	28.6	11.6	20.8	72.3	43.7	33.3
CRST [15]	91.8	49.7	66.1	6.4	14.5	5.2	8.6	21.5	56.3	50.7	30.5	30.7	46.3	34.2	11.7	22.1	72.2	44.9	34.1
SVMin [4]	93.4	53.4	62.2	12.3	15.5	7.0	8.5	18.0	54.3	57.1	31.2	29.6	45.2	35.6	11.5	22.7	72.4	45.5	34.8
CVRN [7]	93.6	52.3	65.3	7.5	15.9	5.2	7.4	22.3	57.8	48.7	32.9	30.9	49.6	38.9	18.0	25.2	72.7	46.7	35.7
UniDAPS [14]	93.9	53.1	63.9	8.7	14.0	3.8	10.0	26.0	53.5	49.6	38.0	35.4	57.5	44.2	28.9	29.8	72.9	49.5	37.6
EDAPS w/ MiT-B2 (Ours)	90.3	64.8	80.0	20.7	**32.0**	47.9	45.4	63.3	85.1	**71.8**	46.8	48.0	64.0	**52.6**	34.1	36.2	78.9	68.7	55.1
EDAPS w/ MiT-B5 (Ours)	91.0	**68.5**	**80.9**	**24.1**	29.0	**50.1**	**47.2**	**67.0**	**85.3**	**71.8**	**50.9**	**51.2**	**64.7**	47.7	**36.9**	**41.5**	**79.2**	**70.5**	**56.7**

Table 2: Comparison with state-of-the-art methods on Cityscapes → Mapillary Vistas benchmark for UDA Panoptic Segmentation. For clarity, per class PQs are reported. The results of EDAPS are averaged over 3 random seeds.

UDA Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	m.bike	bike	mSQ	mRQ	mPQ
CRST [15]	77.0	22.6	40.2	7.8	10.5	5.5	11.3	21.8	56.5	77.6	29.4	18.4	56.0	27.7	11.9	18.4	72.4	39.9	30.8
FDA [13]	74.3	23.4	42.3	9.6	11.2	6.4	15.4	23.5	60.4	78.5	33.9	19.9	52.9	8.4	17.5	16.0	72.3	40.3	30.9
AdvEnt [11]	76.2	20.5	42.6	6.8	9.4	4.6	12.7	24.1	59.9	83.1	34.1	22.9	54.1	16.0	13.5	18.6	72.7	40.3	31.2
CVRN [7]	**77.3**	21.0	47.8	10.5	13.4	7.5	14.1	25.1	62.1	**86.4**	**37.7**	20.4	55.0	21.7	14.3	21.4	73.8	42.8	33.5
EDAPS w/ MiT-B5 (Ours)	58.8	**43.4**	**57.1**	**25.6**	**29.1**	**34.3**	**35.5**	**41.2**	**77.8**	59.1	35.0	**23.8**	**56.7**	**36.0**	**24.3**	**25.5**	**75.9**	**53.4**	**41.2**

Table 3: Performance comparison between adversarial and self training based models (SYNTHIA → Cityscapes).

UDA Method	mAP	mIoU	mSQ	mRQ	mPQ
EDAPS Adversarial Train	23.6	39.4	63.8	35.0	26.2
EDAPS Self Train (Ours)	**34.4**	**57.5**	**72.7**	**53.6**	**41.2**

which makes our architecture study more difficult. To provide a more comprehensive picture, we additionally train EDAPS with adversarial training [10] in Tab. 3. Consistent with UDA semantic segmentation, self-training achieves better results for UDA panoptic segmentation.

5. Ablation Study of Instance Losses

EDAPS uses a top-down instance decoder, which is trained using 5 losses. Even though the effect of these losses is well explored for supervised panoptic segmentation, the influence of the different losses on UDA panoptic segmentation has not been studied so far. Therefore, we present a detailed ablation study analyzing the effect of each instance loss on the domain-adaptive panoptic performance (mPQ). Furthermore, we provide the domain-adaptive instance segmentation performance (mAP), which helps to understand the significance of each instance loss towards the adaptation process for instance segmentation.

We ablate all 5 instance losses, including the losses in the RPN and RoI heads. There are 2 losses in the RPN head, RPN bounding-box classification and regression losses ($\mathcal{L}_{\text{RPN-cls}}$, $\mathcal{L}_{\text{RPN-box}}$), and 3 losses in the RoI head, RoI bounding-box classification, regression, and RoI mask classification losses ($\mathcal{L}_{\text{RoI-cls}}$, $\mathcal{L}_{\text{RoI-box}}$, $\mathcal{L}_{\text{RoI-mask}}$). For this ablation, we train 8 models with different combinations of the instance losses on the SYNTHIA → Cityscapes benchmark. The models are trained following the same setup as EDAPS.

The results of the ablation study in Table 4 provide interesting observations: Without RPN losses, the mPQ decreases from 41.2 to 30.8. At a closer look, we note that instance segmentation (mAP) and recognition quality (mRQ) are adversely affected the most. That implies, in the absence of good quality region proposals, the network struggles to generate correct instance segmentation masks, and there is an increase in false detections (false positives and false negatives). Besides, the RPN box regression loss contributes more to the overall performance improvement than the RPN box classification loss.

In the absence of the RoI head's box classification and regression losses, the model shows the lowest mPQ, mAP, mRQ, and mSQ of 29.5, 0.5, 38.4, 45.5, respectively. It

Table 4: EDAPS instance head losses ablation study on the SYNTHIA → Cityscapes UDA panoptic benchmark. The results of the trained models are averaged over 3 random seeds.

	$\mathcal{L}_{\text{RPN-cls}}$	$\mathcal{L}_{\text{RPN-box}}$	$\mathcal{L}_{\text{RoI-cls}}$	$\mathcal{L}_{\text{RoI-box}}$	$\mathcal{L}_{\text{RoI-mask}}$	mAP	mIoU	mSQ	mRQ	mPQ
Model 1			✓	✓	✓	9.3 $\pm$4.4	57.5 $\pm$0.4	70.9 $\pm$7.3	43.2 $\pm$5.8	30.8 $\pm$1.3
Model 2	✓		✓	✓	✓	4.8 $\pm$2.1	57.9 $\pm$1.1	62.9 $\pm$2.4	38.8 $\pm$0.3	29.6 $\pm$0.2
Model 3		✓	✓	✓	✓	16.9 $\pm$5.9	57.8 $\pm$0.4	72.5 $\pm$1.6	45.1 $\pm$3.2	34.7 $\pm$2.4
Model 4	✓	✓			✓	0.5 $\pm$0.3	57.3 $\pm$0.4	45.5 $\pm$0.1	38.4 $\pm$0.7	29.5 $\pm$0.5
Model 5	✓	✓		✓	✓	2.3 $\pm$2.0	57.9 $\pm$0.5	45.6 $\pm$0.1	38.4 $\pm$0.4	29.5 $\pm$0.3
Model 6	✓	✓	✓		✓	29.6 $\pm$0.4	57.5 $\pm$0.4	71.7 $\pm$0.5	50.4 $\pm$0.7	38.2 $\pm$0.7
Model 7	✓	✓	✓	✓		9.7 $\pm$1.9	57.0 $\pm$1.3	65.3 $\pm$3.8	43.7 $\pm$1.6	32.7 $\pm$0.9
Model 8	✓	✓	✓	✓	✓	34.4 $\pm$0.5	57.5 $\pm$0.0	72.7 $\pm$0.2	53.6 $\pm$0.5	41.2 $\pm$0.4

implies that the RoI-pooled features play a vital role; the instance head trained without losses on the RoI features struggles to achieve high-quality instance segmentation. Interestingly, the RoI head's box classification loss contributes more to the overall performance gain than the box regression loss. Since the RPN box regression loss already helps the network to learn better instance bounding boxes, even if the RoI head box regression loss is turned off, it achieves an mPQ of 38.2, which is already better than the 32.1 mPQ of the prior work CVRN [7]. However, it is crucial for the RoI head to learn the correct box label classification; since the RPN box classification loss is only responsible for providing correct binary labels (object vs. no-object) for the region proposals, the RoI box classification loss helps the instance head to learn correct instance class labels (i.e., the 8 thing object classes) for the RoI-predicted boxes. Finally, in the absence of the RoI mask classification loss, the mPQ goes down from 41.2 to 32.7, which shows that it is crucial for the network to learn the correct binary instance masks to achieve better panoptic segmentation quality.

6. Qualitative Analysis

In this section, we provide additional qualitative prediction results for a visual comparison of the proposed EDAPS and the prior art CVRN [7]. The visualizations for models trained on SYNTHIA → Cityscapes are presented in Fig. 1-4. The major improvements come from better panoptic segmentation of the thing classes *person* (Fig. 1), *rider* (Fig. 2), *car* (Fig. 3); and stuff classes *traffic light*, *traffic sign*, *pole* (Fig. 2, 3, and 4) across different object scales, appearance, and viewing angles. In general, EDAPS can better delineate object boundaries, resulting in better-quality pixel-level panoptic segmentation. Note that the detected object shapes (e.g., *person*, *rider*, *car*) predicted by the EDAPS resemble more real-world object shapes when compared to CVRN [7]. Thanks to the domain-robust Mix Transformer (MiT-B5) [12] backbone, EDAPS can learn a richer set of domain-invariant semantic and instance features helpful in better segmentation of fine structures. EDAPS can better segment the occluded object instances in a crowded scene such as *person* (Fig. 1 row 1-5), *rider* (Fig. 2 row 1), *car* (Fig. 3 row 1-8). Moreover, the *person* segments predicted by EDAPS preserve finer details of the human body even when instances are occluded. Similar observations can be made for the *rider* and *car* classes. For large object instances (such as *bus*), EDAPS can segment out the entire object, whereas CVRN fails to do so (Fig. 2 row 8, Fig. 4 row 1). EDAPS can provide better segmentation for the *traffic light* (Fig. 2 row 1, 8; Fig. 4 row 3, 4), and *traffic sign* (Fig. 2 row 4, 8; Fig. 4 row 1, 4, 5).

In addition, we show visual qualitative results on SYNTHIA → Mapillary Vistas UDA panoptic benchmark (Fig. 5-7). EDAPS segments better the *pole* instance (Fig. 5 row 5). In Fig. 6 and 7, we present a visual comparison with the Source-Only model. It can be observed that the Source-Only model struggles to learn the correct class labels and instance masks, whereas EDAPS successfully bridges the domain gap by learning the correct semantics and instances. EDAPS produces better panoptic segmentation for the *bus* (Fig. 6 row 1, Fig. 7 row 1, 2, 3), *rider* (Fig. 6 row 2, 3, Fig. 7 row 5), *motorbike* (Fig. 6 row 3, Fig. 7 row 6), *car* (Fig. 7 row 3, 4), *traffic sign* (Fig. 6 row 6, Fig. 7 row 1). Finally, the visual predictions on Cityscapes → Foggy Cityscapes are shown in Fig. 8.

7. Visual Comparison: EDAPS vs. M-Dec-BU

This section offers a visual comparison of the predictions made by EDAPS and M-Dec-BU on the SYNTHIA → Cityscapes benchmark, as depicted in Fig. 9. We observed that the M-Dec-BU baseline model tends to segment objects (like pedestrians, cars, buses, and riders) into smaller parts than necessary (i.e., over-segmentation). Notice that the pedestrian, car, and bus instances in Figs. 9 (a-d) are over-segmented. This over-segmentation problem is more prominent in scenes with large and occluded objects.

The M-Dec-BU model adopts a bottom-up approach for instance segmentation [3]. Unlike top-down methods [5], M-Dec-BU's instance head does not directly predict instance segmentation masks. Rather, it predicts instance centers and offsets. An additional post-processing step is required to generate the class-agnostic instance segmentation masks from these predicted centers and offsets. We found that the center predictions are not sufficiently robust under a domain shift (even with domain adaptation) to support reliable post-processing on the target domain which leads to an over-segmentation problem as discussed above. In contrast, we noticed that EDAPS's top-down instance segmentation head predicts highly generalizable instance masks on the target domain resulting an improved instance segmentation performance (mAP 34.4%) as compared to 17.6% mAP of M-Dec-BU.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2

[2] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2

[3] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020. 1, 4

[4] Dayan Guan, Jiaxing Huang, Shijian Lu, and Aoran Xiao. Scale variance minimization for unsupervised domain adaptation in image segmentation. *Pattern Recognition*, 112:107764, 2021. 2

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[6] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 1

[7] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Cross-view regularization for domain adaptive panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10133–10144, 2021. 1, 2, 3, 5, 6, 7, 8, 9

[8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 1

[9] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 687–704, 2018. 1

[10] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 2

[11] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 2

[12] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 3

[13] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 2

[14] Jingyi Zhang, Jiaxing Huang, and Shijian Lu. Hierarchical mask calibration for unified domain adaptive panoptic segmentation. *arXiv preprint arXiv:2206.15083*, 2022. 2

[15] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 2

Figure 1: Example predictions showing better panoptic segmentation for *person* on SYNTHIA → Cityscapes.

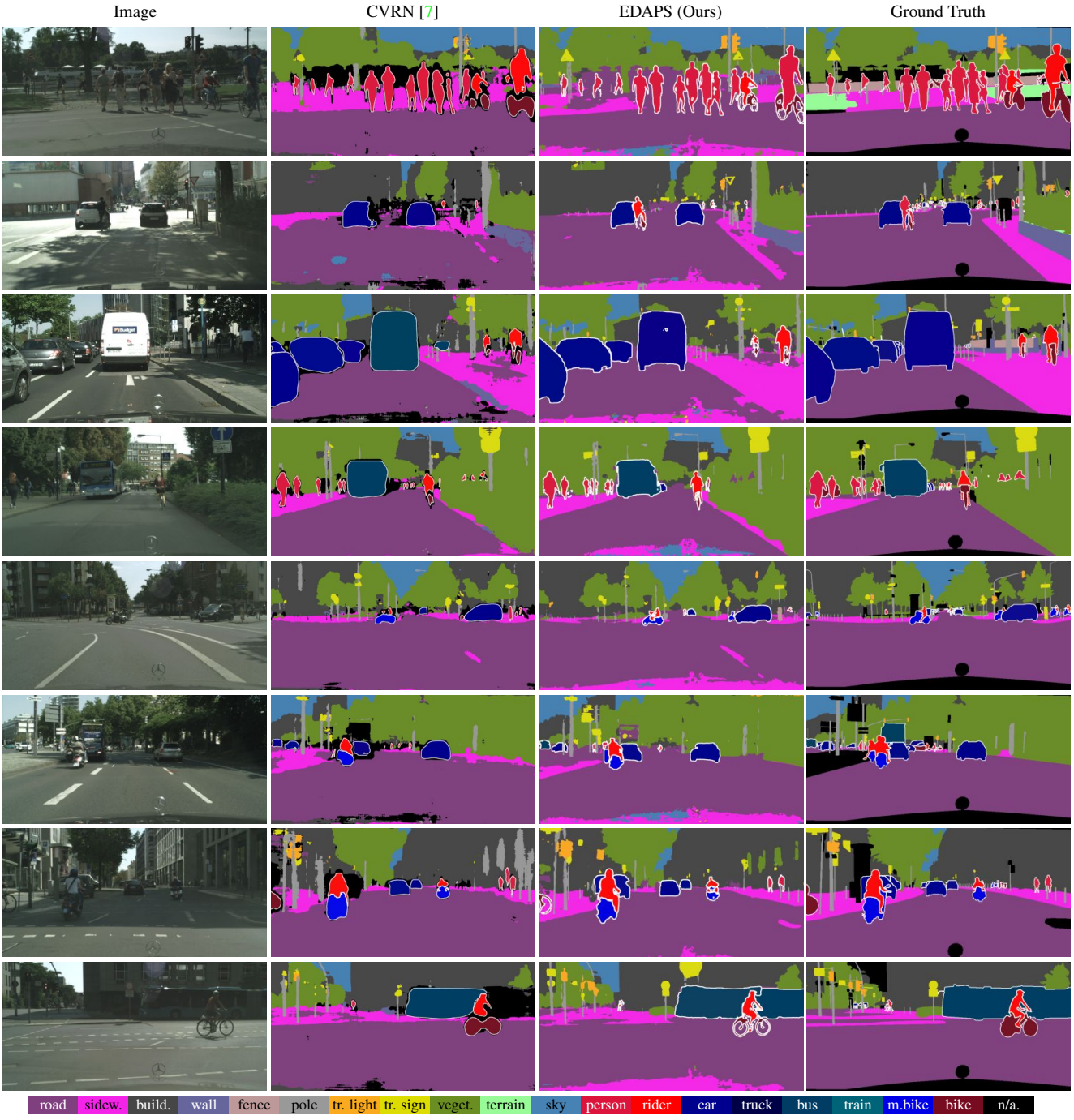

Figure 2: Example predictions showing better panoptic segmentation for *rider*, *motorbike*, *bus*, and *sign* classes on SYNTHIA → Cityscapes.

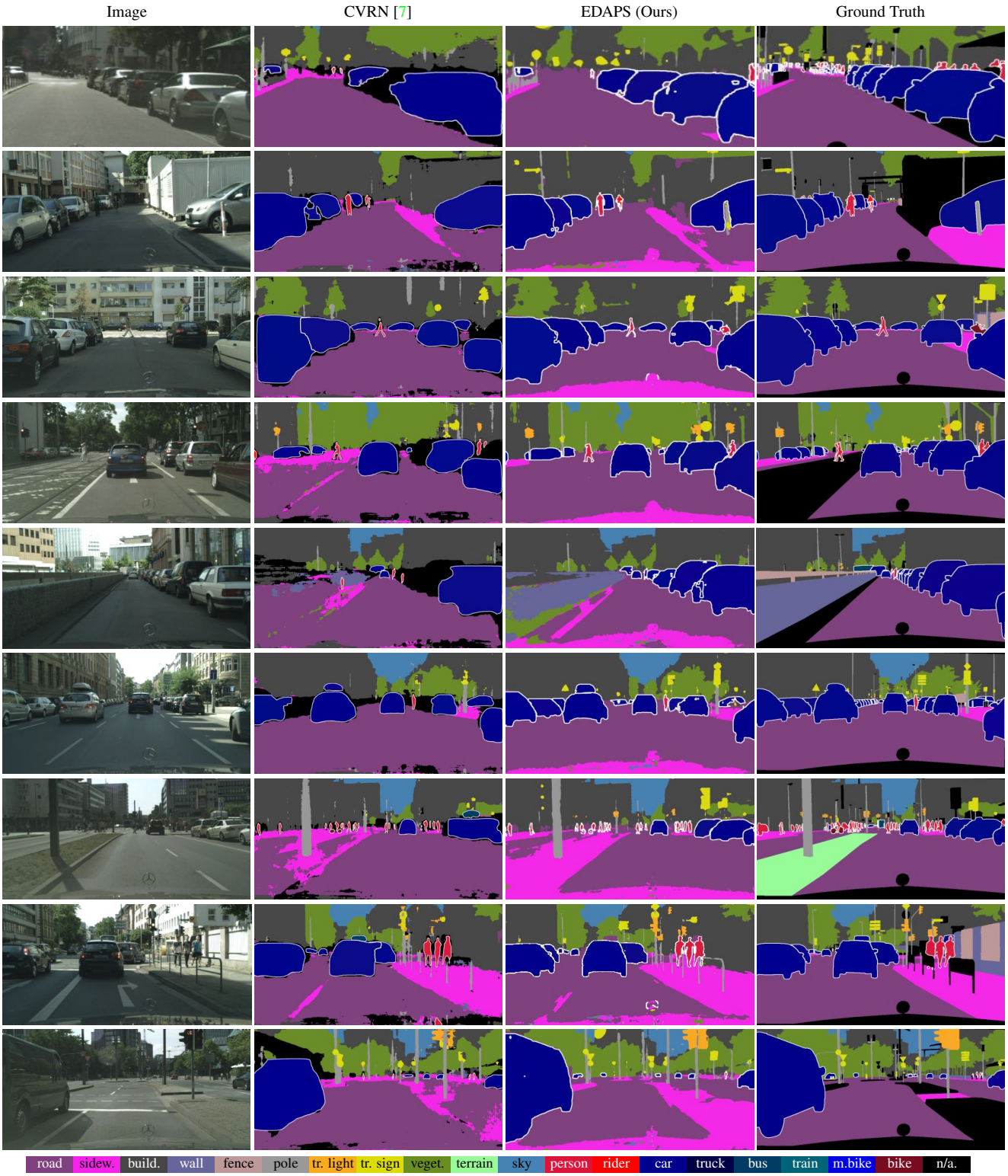

Figure 3: Example predictions showing better panoptic segmentation for thing (*car*) and stuff (*wall*, *sign*, *light*) classes on SYNTHIA → Cityscapes.

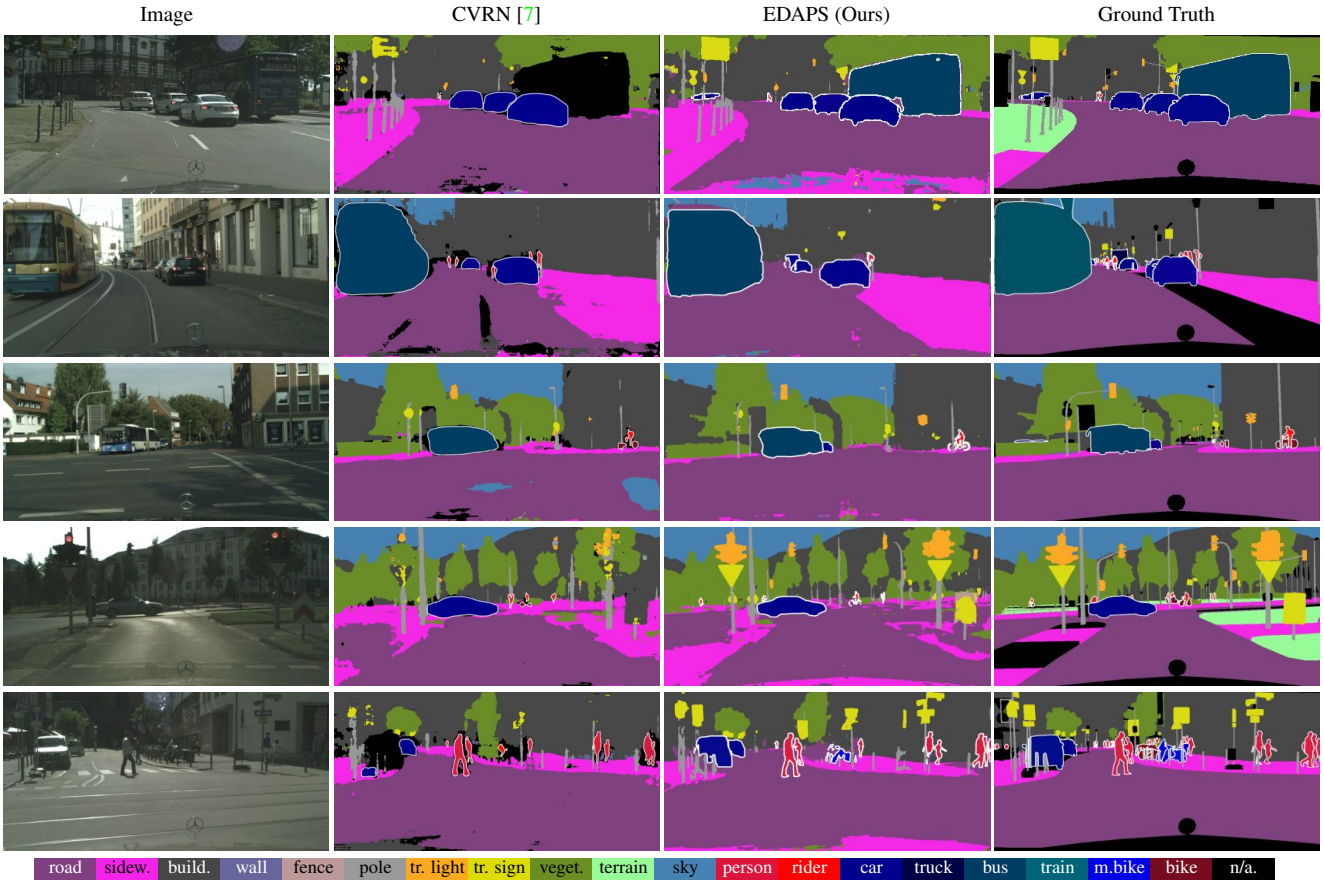

Figure 4: Example predictions showing better panoptic segmentation for *bus*, *traffic sign* and *traffic light* on SYNTHIA → Cityscapes.

Figure 5: Example predictions on SYNTHIA → Mapillary Vistas.

Figure 6: Example predictions on SYNTHIA → Mapillary Vistas.

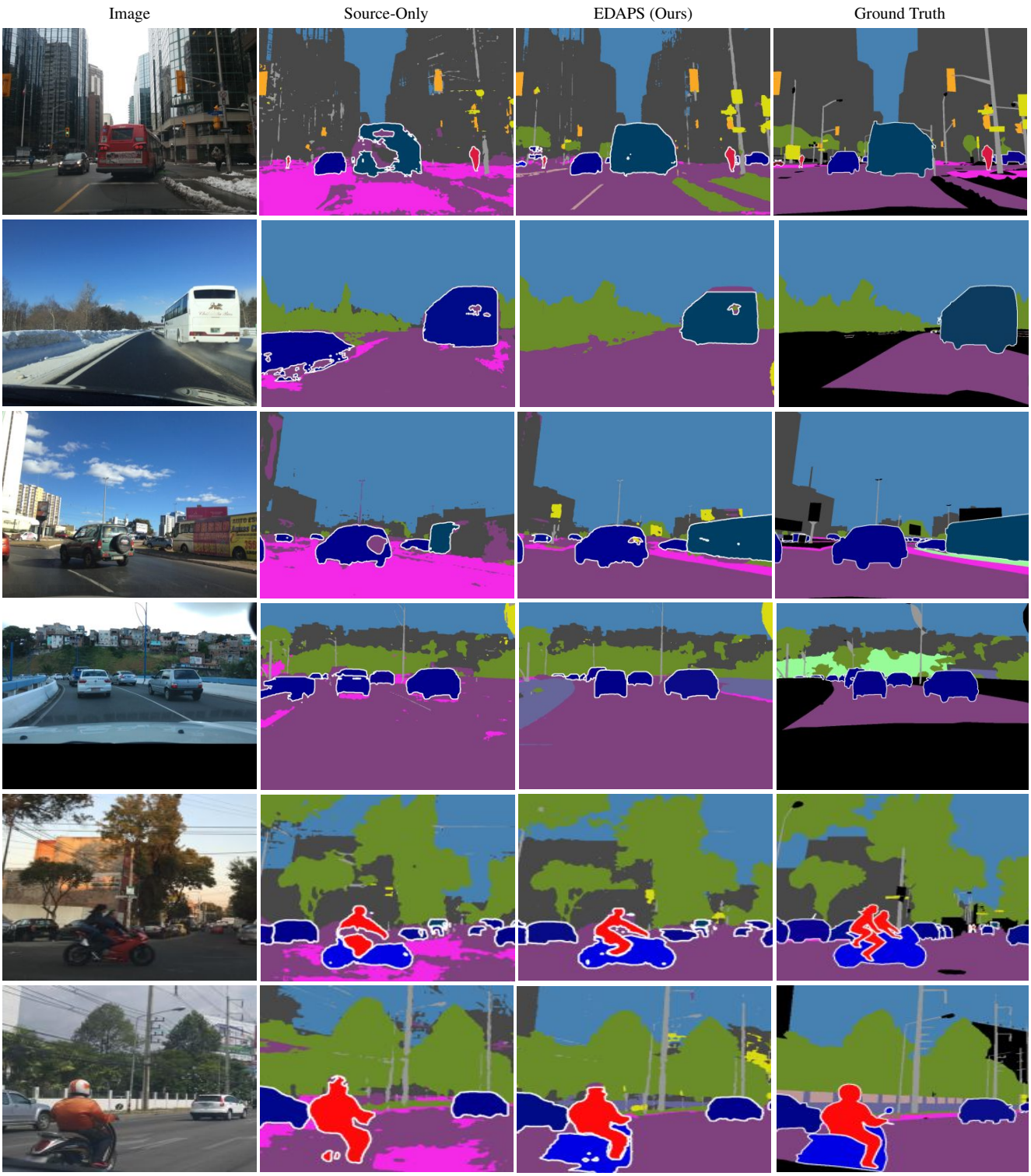

Figure 7: Example predictions on SYNTHIA → Mapillary Vistas.

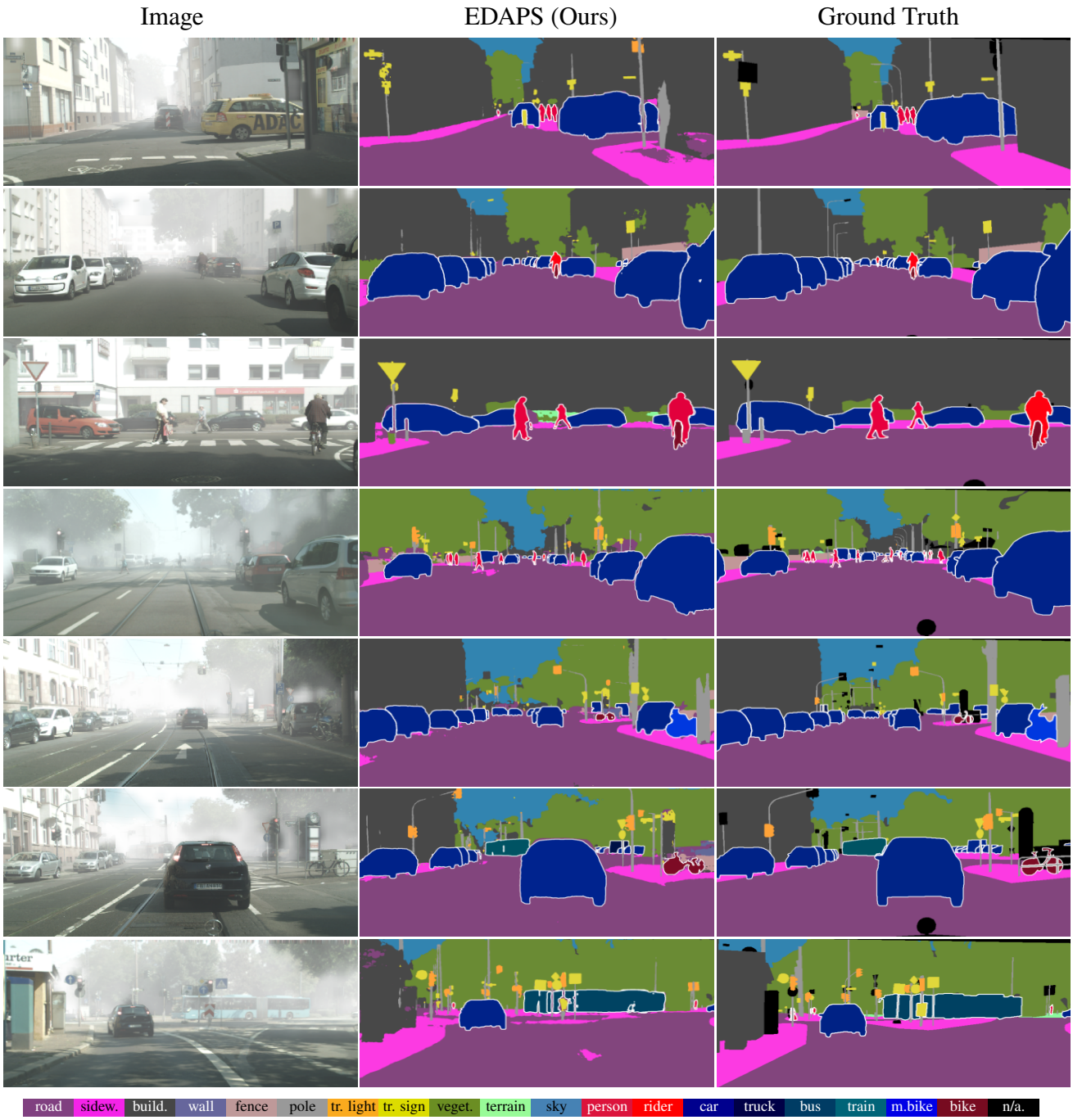

Figure 8: Visual prediction results of EDAPS on Cityscapes → Foggy Cityscapes.

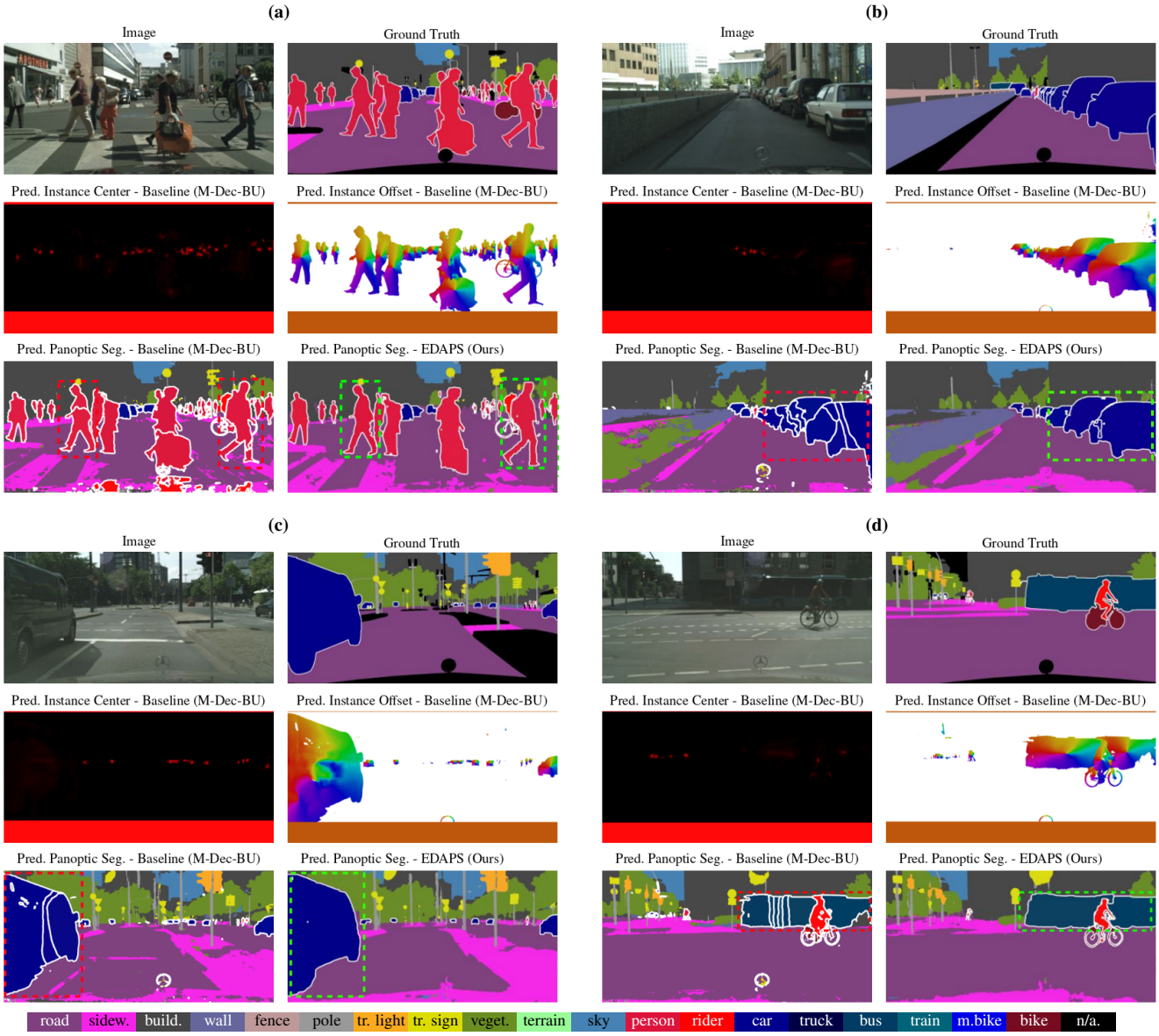

Figure 9: Visual comparison of EDAPS and M-Dec-BU (baseline) predictions on SYNTHIA → Cityscapes.