Pin the Memory: Learning to Generalize Semantic Segmentation -Supplementary Materials -

Jin Kim[1] Jiyoung Lee[2] Jungin Park[1] Dongbo Min[3*] Kwanghoon Sohn[1*]
[1]Yonsei University [2]NAVER AI Lab [3]Ewha Womans University
{kimjin928, newrun, khsohn}@yonsei.ac.kr lee.j@navercorp.com dbmin@ewha.ac.kr

In this document, we describe second-order gradient flow of our method and details of experiments, and provide additional ablation study for analysis of memory update. Moreover, we complement qualitative and quantitative comparisons to state-of-the-art methods.

A. Second-Order Gradient Flow

In Fig. 1, we depict the gradient flow of the optimization in the meta-testing step. In this process, we compute the gradient of the original parameters $\{\Theta\}_{E,U,D}$ for the meta-testing loss and generate the second-order gradients by differentiating the parameters $\{\Theta\}'_{E,U,D}$ used in the meta-testing step with the original parameters. These second-order gradients make the original parameters learn to (1) write the domain-independent features to the current memory $\mathcal{M}$ from the meta-train image and (2) ensure the generalization ability of the memory-guided feature for the meta-test image.

B. Implementation Details

B.1. Data Split and Augmentation

The batch size per domain was 4 for multi-source domain training and 8 for single-source domain training. Following the setting from RobustNet [1], standard augmentations such as color jittering (brightness of 0.4, contrast of 0.4, saturation of 0.4, and hue of 0.1), Gaussian blur, random cropping, random horizontal flipping, and random scaling with the range of [0.5, 2.0] were conducted to prevent the model from overfitting. To create an artificial domain shift even in a single source domain generalization setting, we applied higher intensity random color jittering (brightness of 0.8, contrast of 0.8, saturation of 0.8, and hue of 0.3) and Gaussian blur only to the images used in the meta-testing step.

B.2. Training and Optimization

We implemented our approach with PyTorch and conducted experiments by adopting DeepLabV3+ [2] with ResNet-50 [3] backbone network. The output stride of DeepLabV3+ was set to 16 and adopted the auxiliary per-pixel cross-entropy loss proposed in PSPNet [4] with a coefficient of 0.4 to make a fair comparison with the normalization based DG method [1]. We performed memory operation using the feature map of 256 channel dimensions after the ASPP [2] module to leverage the multiple receptive fields and reduce GPU memory usage. We also adopted DeepLabV2 [5] with ResNet-101 for a fair comparison with multi-source unsupervised domain adaptation methods. For all the experiment, we initialized backbones with ImageNet [6] pre-trained model. The optimizer was SGD with momentum of 0.9. The learning rate of the meta-testing step β was 1e-2 initially and decreased with exponential learning rate policy with the gamma of 9. The learning rate of the meta-training step α was set to 1/4 of the outer learning rate β to stabilize the gradient-based meta optimization [7,8]. We set the maximum iterations to 120K but early stop at 30K iterations, except for ResNet-101 models trained for 70K. The coefficients of memory divergence loss and feature cohesion loss, λ_1 and λ_2, was set to 0.02 and 0.2, respectively.

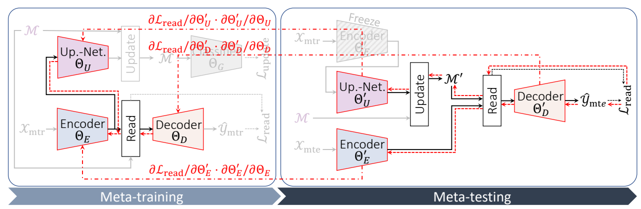


Figure 1. Illustration of the gradient flow (red dotted lines) in the optimization of meta-testing step.

B.3. Re-implemented Methods

While IBN-Net [9] improved generalization ability by mixing instance normalization and batch normalization in the backbone, RobustNet [1] previously have shown SOTA performance by selectively removing the channel covariance of the backbone. We re-implemented these two methods by setting the hyper-parameters according to the public code by RobustNet [1][1]. To verify the effective-

*Corresponding authors.

[1]https://github.com/shachoi/RobustNet

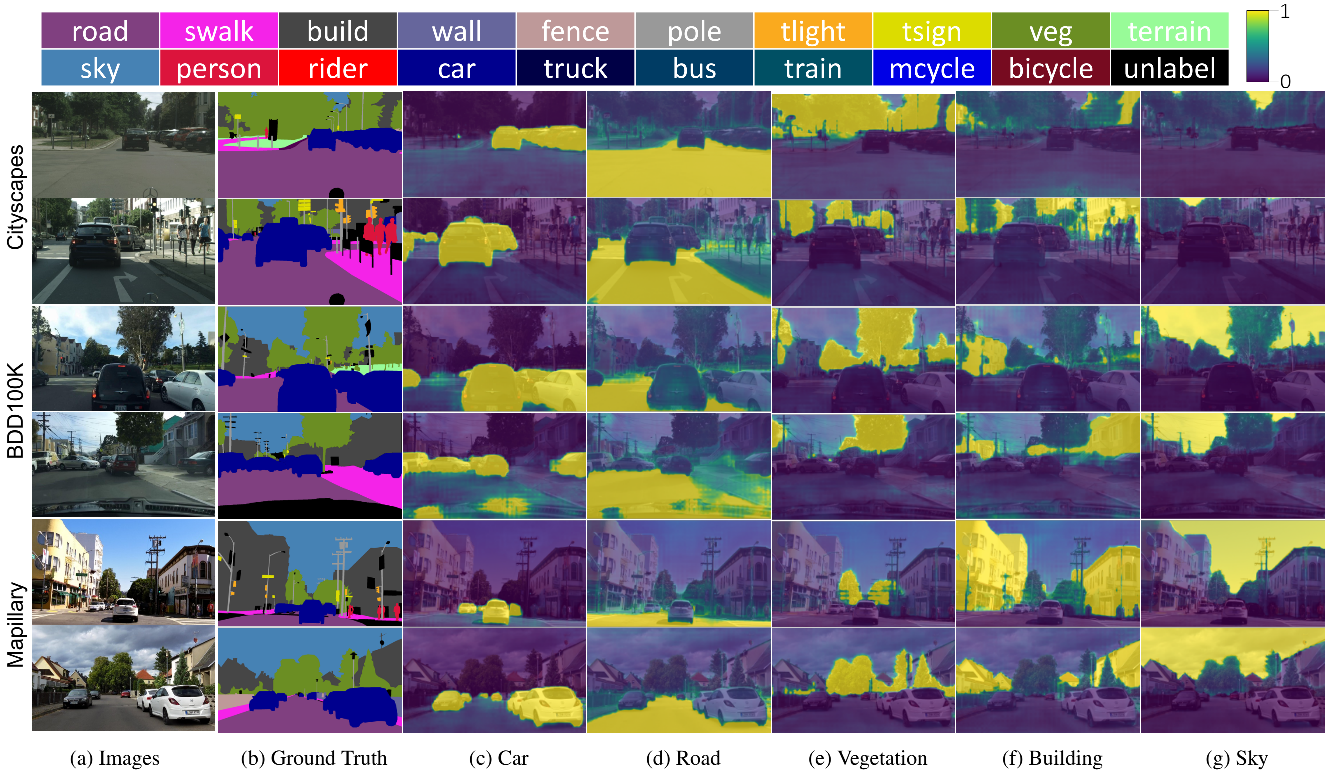


Figure 2. **Source (G+S)→Target (C, B, M):** Visualization of the memory weights for each class on the Cityscapes, BDD100K and Mapillary dataset. We adopt DeepLabV3+ with ResNet50.

ness of our memory-guided meta-learning method, we re-implemented the MLDG [10] which is meta-learning based DG method. The augmentations and learning rates of MLDG were same with our method. Recently, TSMLDG [11] purely uses meta-learning for DG and proposes a method for target-domain batch normalization on test-time. We re-implemented TSMLDG by setting the test-batch size to 4 and updating batch statistics of the MLDG model in testing time on the unseen target domain according to the code of TSMLDG[2].

C. Additional Results

C.1. Ablation Study

Analysis of memory updating network. To verify the effectiveness of the memory updating network, we conduct an ablation study about memory updating network. In Table 1, we can observe that the memory updating network has notable contribution to the performance gain for all datasets by storing generalizable features into the memory.

More visualization of memory activation. To complement the Fig. 6 of the main paper, we additionally visualize the memory weight for the input image from all the unseen datasets in Fig. 2. Regardless of the environment, the memory corresponding to each object category is well activated, so that the feature of the pixel can receive a guide of the appropriate memory feature. In addition, the results demonstrate that our memory item contains the generic features of the categories, even though the memory has been trained on synthetic datasets.

[2]https://github.com/koncle/TSMLDG

Memory Update Net.	Cityscapes	BDD100K	Mapillary	Avg.
✗	41.28	37.25	40.64	39.72
✓	**44.51**	**38.07**	**42.70**	**41.76**

Table 1. **Source (G+S)→Target (C, B, M):** Performance with or without memory updating network.

Methods	$\mathcal{L}_{seg}$	$\mathcal{L}_{coh}$	$\mathcal{L}_{div}$	Cityscapes	BDD100K	Mapillary	Avg.
IBN-Net [9]	✓	✗	✗	35.55	32.18	38.09	35.27
MLDG [10]	✓	✗	✗	38.84	31.95	35.60	35.46
Ours	✓	✗	✗	38.22	33.12	37.10	36.15
	✓	✓	✓	**44.51**	**38.07**	**42.70**	**41.76**

Table 2. **Source (G+S)→Target (C, B, M):** Mean IoU(%) comparison between the DG methods with only standard segmentation loss, $\mathcal{L}_{seg}$. All networks are DeepLabV3+ with ResNet50.

Loss comparison with previous works. To convincingly compare our proposed losses with previous works, we re-implemented our model using only standard loss (cross entropy) in Table 2. Without the proposed losses, our method

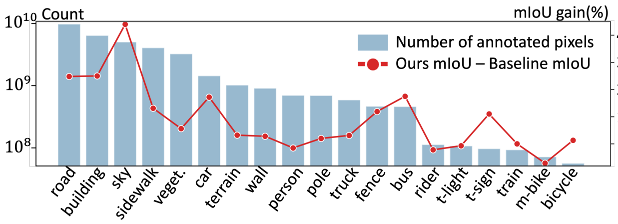


Figure 3. The correlation between the number of pixels per class in source datasets (G, S) and performance gain on BDD100K dataset.

Methods	BDD100K	Mapillary	Avg.
MCIBI [12]	41.65	50.18	45.92
Ours	**46.78**	**55.10**	**50.94**

Table 3. **Source (C)→Target (B, M):** Mean IoU(%) comparison with MCIBI [12]. All networks are DeepLabV3 with ResNet50.

still shows competitive performance against IBN-Net [9] and MLDG [10] due to the help of memory items. Moreover, $\mathcal{L}_{\text{coh}}$ and $\mathcal{L}_{\text{div}}$ lead to substantial performance gain by facilitating the effective memory read/update procedures in training.

Correlation between performance gain and class distribution. The generalization capability usually benefits from the diversity and amount of the training samples. However, the data imbalance between classes in current benchmarks is significant since the different occurrence frequency and variants of shape among classes. In Fig. 3, we analyze the correlation between the performance gain over the baseline in Table 1 of the main paper and the number of training samples. While the high mIoU gain is attained for the class (e.g. road, building, sky) with sufficient training samples, it becomes lower for some minor classes. We remain this problem due to the limitation of current benchmarks as future work.

Comparison with MCIBI. We conduct comparison with MCIBI [12] which is a memory network designed for conducting semantic segmentation on seen domain dataset. To compare generalization performance, we used the author-provided MCIBI model pre-trained on Cityscapes and evaluated on the other real datasets regarding single-source setting. In Table 3, we can see that our memory module outperforms MCIBI on unseen domain datasets. It thus points out that using our non-parametric memory loss and leveraging meta-learning to store shared information among the same class play important roles in improving generalization capability of the segmentation network.

C.2. Full Comparison with State-Of-The-Art.

Quantitative results. Table 4 shows the results evaluated on the real datasets with various segmentation models regarding to single-source domain generalization setting. Even though the networks were trained on the GTAV dataset only,

Backbone	Methods	Seg. model	Cityscapes	BDD100K	Mapillary
Resnet50	Baseline	FCN-8s	32.50	26.70	25.70
	DRPC [13]		37.40	32.10	34.10
	Baseline†	DeepLabV3+	29.00	25.10	28.20
	IBN-Net† [9]		33.90	32.30	37.80
	RobustNet† [1]		36.60	35.20	40.30
	Baseline		31.60	26.70	29.00
	MLDG‡ [10]		36.70	32.10	32.20
	Ours		41.00	34.60	37.40
Resnet101	FSDR [14]	DeepLabV2	44.75	39.66	40.87
	Ours		**44.90**	**39.71**	**41.31**

Table 4. **Source (G)→Target (C, B, M):** Mean IoU(%) comparison of other SOTA methods using various segmentation models and backbones. MLDG [10] is re-implemented. Results with † are from [1].

our method obtained the best generalization performance on the Cityscapes dataset. Our method also achieved a relatively high-performance gain over our baseline results on the BDD100K and Mapillary datasets. We also compare with the performance of FSDR [14] where we used the author-provided model parameters of FSDR pre-trained on GTAV. Our model performs better than FSDR on all the target domain datasets.

Furthermore, we report the per-class IoU scores for Table 2 and Table 4 of the main paper in Table 5 and Table 6, respectively. Table 5 shows the performance of Cityscapes, BDD100K, and Mapillary with DG models trained on GTA5, Synthia, and IDD datasets. The results show that our method increased the average performance of each class without overfitting a specific category in the unseen domain. In Table 6, we compare the performance on the real-world datasets with state-of-the-art multi-source UDA methods that leverage target domain images on training time. Although UDA is a much easier setting than domain generalization, our DG method achieved the highest performance on the BDD100K and competitive results on the Cityscapes.

Qualitative results. To qualitatively describe the superiority of our method, we compare the segmentation results with other state-of-the-art DG methods. We trained all DG methods on multi-source synthetic datasets (*i.e.* GTAV [20] and Synthia [21]), and tested on the *unseen* real-world datasets [22–24].

In Fig. 4, we firstly conduct an additional comparison of the segmentation results on the Cityscapes [22] dataset. The baseline model showed weakness to changes in brightness due to shadows or changes in places such as side streets and parking lots in the real world. In addition, results from all the other methods were damaged to predict objects such as trains or trucks in the real world. In contrast, our method predicted road, train, truck, and vegetation relatively well, showing robustness to domain change.

Fig. 5 and Fig. 6 show the predicted segmentation results on the BDD100K dataset. Compared to the Cityscapes dataset that only contains images acquired primarily in daytime and relatively simple weather conditions (*i.e.* overcast or

	Methods	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegettion	terrain	sky	person	rider	car	truck	bus	train	m-bike	bicycle	mIoU(%)
Cityscapes	Baseline	88.6	45.9	85.5	38.2	29.7	46.0	**45.0**	41.6	88.6	**43.3**	**93.2**	73.5	44.0	81.4	46.3	29.3	0.3	30.0	47.3	52.5
	IBN-Net‡ [9]	90.2	52.0	86.9	38.4	31.8	47.8	43.6	43.8	**89.3**	42.3	91.9	72.8	42.8	82.3	50.5	48.6	0.2	28.8	49.3	54.4
	RobustNet‡ [1]	90.4	48.1	86.8	36.1	34.6	47.3	39.3	**43.9**	89.2	40.7	92.1	73.2	44.6	87.8	51.7	50.8	0.0	32.2	50.6	54.7
	MLDG‡ [10]	91.2	50.8	87.4	39.5	30.4	**49.0**	39.4	42.5	89.1	39.2	93.0	**74.1**	**46.0**	86.4	50.3	49.6	0.6	31.4	50.5	54.8
	TSMLDG‡ [11]	**92.1**	**52.7**	87.4	37.1	31.3	48.5	40.5	42.7	89.1	39.2	92.6	72.1	41.8	89.0	49.3	47.2	0.6	18.5	35.8	53.0
	Ours	91.0	51.6	**87.9**	**43.1**	**36.6**	47.6	38.7	43.1	**89.3**	41.8	93.0	73.9	41.9	**89.1**	**58.9**	**55.8**	**2.0**	**37.2**	**52.5**	**56.6**
BDD100k	Baseline	89.8	42.7	76.8	14.1	41.9	43.6	34.7	31.7	81.0	40.6	90.3	62.2	26.4	82.2	26.7	40.2	0.0	38.1	38.8	47.5
	IBN-Net‡ [9]	88.5	46.7	**78.7**	**20.6**	40.8	**45.4**	**39.4**	32.8	**82.8**	42.1	**91.6**	61.3	21.7	80.7	33.7	**59.8**	0.0	23.4	39.4	48.9
	RobustNet‡ [1]	90.3	42.6	77.7	20.4	39.9	44.6	36.6	33.3	**82.8**	**43.8**	90.8	61.6	21.7	**84.2**	32.3	57.7	0.0	24.8	**46.2**	49.0
	MLDG‡ [10]	90.0	45.7	75.8	15.1	**43.6**	43.1	36.4	32.0	82.3	41.2	89.8	61.1	19.5	80.9	33.4	52.1	0.0	39.5	40.4	48.5
	TSMLDG‡ [11]	**90.8**	45.4	78.0	16.4	34.9	44.5	38.2	**34.7**	81.7	37.3	91.4	57.6	12.9	84.1	**34.3**	53.8	0.0	9.0	36.9	46.4
	Ours	90.4	**52.5**	75.2	18.2	41.8	43.9	38.6	34.4	82.5	40.0	89.7	**62.5**	**26.5**	83.3	31.0	56.2	0.0	**46.2**	40.5	**50.2**
Mapillary	Baseline	87.8	40.3	81.2	29.2	37.9	51.5	42.6	63.7	87.2	48.4	97.2	71.4	44.9	85.9	50.7	30.9	0.5	47.5	40.5	54.7
	IBN-Net‡ [9]	88.5	44.9	**83.6**	35.3	38.3	**53.1**	43.7	63.4	87.5	47.8	**97.4**	71.6	48.3	86.1	47.8	41.0	3.9	45.8	37.1	56.1
	RobustNet‡ [1]	88.2	43.5	83.1	34.2	39.4	52.5	40.2	62.6	87.3	48.4	97.3	72.3	**51.8**	87.7	48.7	51.7	**7.3**	45.4	39.8	56.9
	MLDG‡ [10]	88.0	39.0	82.9	36.6	40.3	51.6	41.7	64.4	87.6	45.7	96.9	**73.0**	51.6	87.3	39.0	44.3	3.5	48.5	41.0	55.9
	TSMLDG‡ [11]	86.1	45.7	79.2	31.4	39.9	52.2	**44.4**	61.8	84.2	38.5	88.1	68.8	49.2	86.6	31.0	31.8	5.3	42.7	35.3	52.7
	Ours	**89.2**	**48.1**	83.2	**36.9**	**40.6**	52.4	42.3	**64.8**	**87.7**	**49.6**	97.3	72.2	47.3	**89.2**	**53.6**	**55.9**	3.9	**49.4**	**44.2**	**58.3**

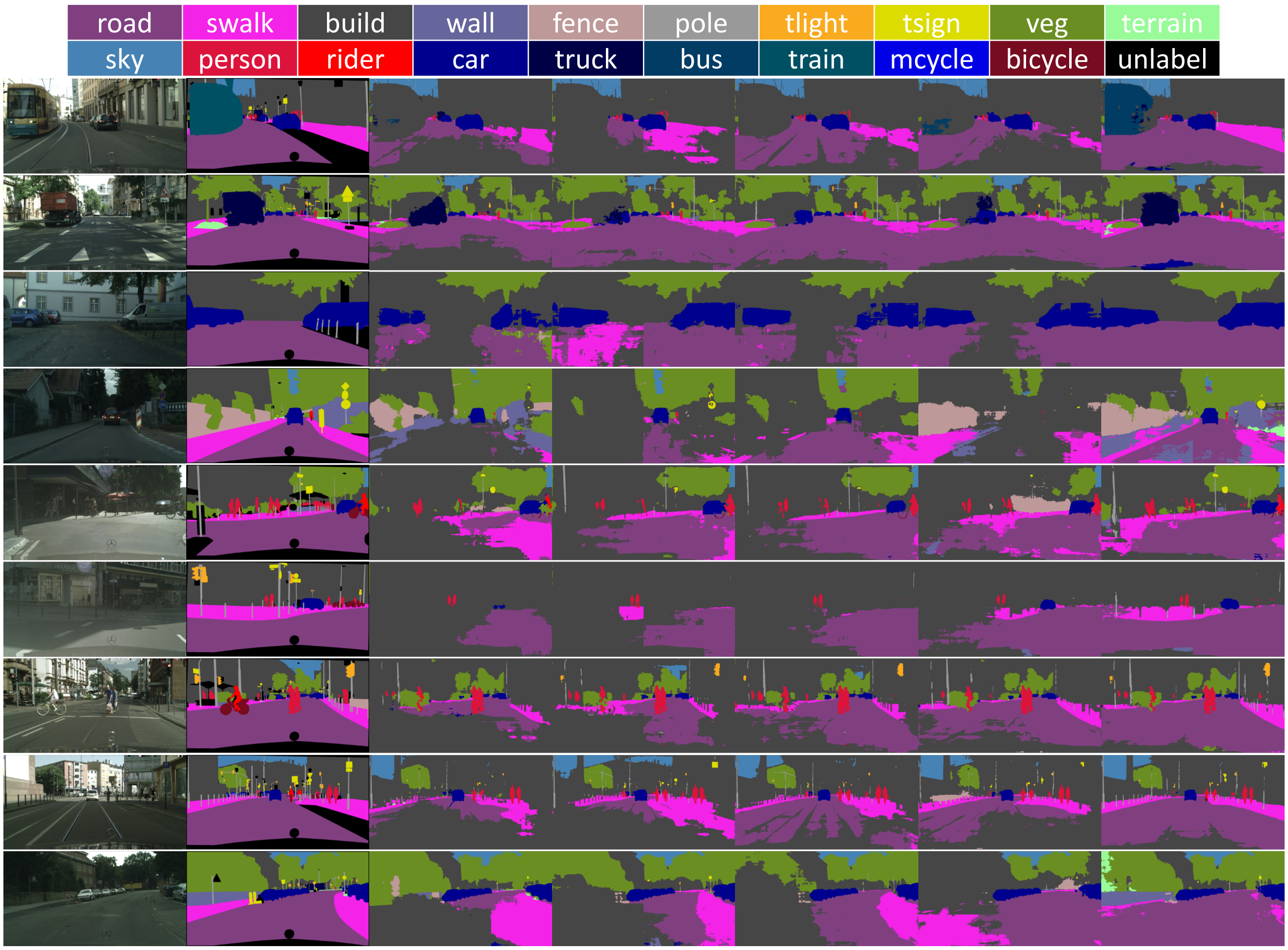
Table 5. **Source (G+S+I)→Target (C, B, M):** Mean IoU(%) and per-class IoU(%) comparison of other state-of-the-art DG methods for semantic segmentation. We re-implemented all methods using DeepLabV3+ with ResNet50 backbone. We re-implement other methods and mark them with ‡.

	Methods	w/Target	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegettion	sky	person	rider	car	bus	m-bike	bicycle	mIoU(%)
Cityscapes	Baseline	✗	77.1	32.4	75.3	13.8	11.5	29.0	13.7	10.3	81.5	79.1	53.1	10.2	80.2	39.0	21.9	11.5	40.0
	CyCADA† [15]	✓	86.8	41.4	74.7	15.5	3.4	27.3	3.8	0.2	73.2	72.4	51.9	12.7	82.7	41.8	18.5	23.3	39.3
	MDAN† [16]	✓	80.6	34.4	73.9	15.9	1.9	22.9	0.1	0.0	73.6	58.9	48.4	12.2	78.8	36.8	14.2	23.7	36.0
	MADAN† [17]	✓	88.1	46.1	79.9	26.4	7.4	30.6	19.0	19.9	80.4	75.9	55.6	15.6	84.1	**47.0**	23.3	26.3	45.4
	MADAN+† [18]	✓	**90.9**	**49.7**	64.9	24.6	13.0	**39.2**	40.0	21.4	80.2	**86.1**	57.3	**25.0**	84.7	35.7	25.2	**38.2**	48.5
	CLSS [19]	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**54.0**
	Ours	✗	87.4	42.7	**82.6**	**29.9**	**21.5**	**39.2**	**48.5**	**34.2**	**85.2**	71.8	**66.6**	17.6	**88.8**	21.5	**26.0**	26.5	49.4
BDD100K	Baseline	✗	55.3	20.9	73.9	15.9	18.9	29.9	11.3	11.9	79.7	76.2	54.7	10.3	79.7	29.3	17.2	14.1	37.4
	CyCADA† [15]	✓	64.9	33.6	73.3	15.8	15.3	29.2	15.9	21.4	79.3	79.0	52.0	12.7	49.7	14.0	17.5	22.5	37.2
	MDAN† [16]	✓	57.6	31.2	53.5	6.5	0.6	20.3	0.0	0.0	73.0	61.7	40.9	9.8	60.4	29.2	10.3	15.6	29.4
	MADAN† [17]	✓	74.5	32.4	71.3	16.5	16.3	30.6	15.1	25.1	**80.6**	78.7	52.2	12.4	70.5	34.0	18.4	19.4	40.4
	MADAN+† [18]	✓	**87.8**	**44.2**	**78.6**	**22.4**	6.8	29.1	11.5	5.3	79.6	74.6	53.6	14.6	**83.0**	**43.4**	19.1	**30.2**	42.7
	Ours	✗	84.5	39.8	69.7	9.0	**26.3**	**36.1**	**43.3**	**31.3**	73.5	**87.1**	**59.2**	**25.5**	81.9	6.6	**38.3**	15.2	**45.5**

Table 6. **Source (G+S)→Target (C, B):** Mean IoU(%) and per-class IoU(%) comparison of other multi-source UDA methods. The segmentation models are all DeepLabV2 with ResNet101. Results with † are from [18].

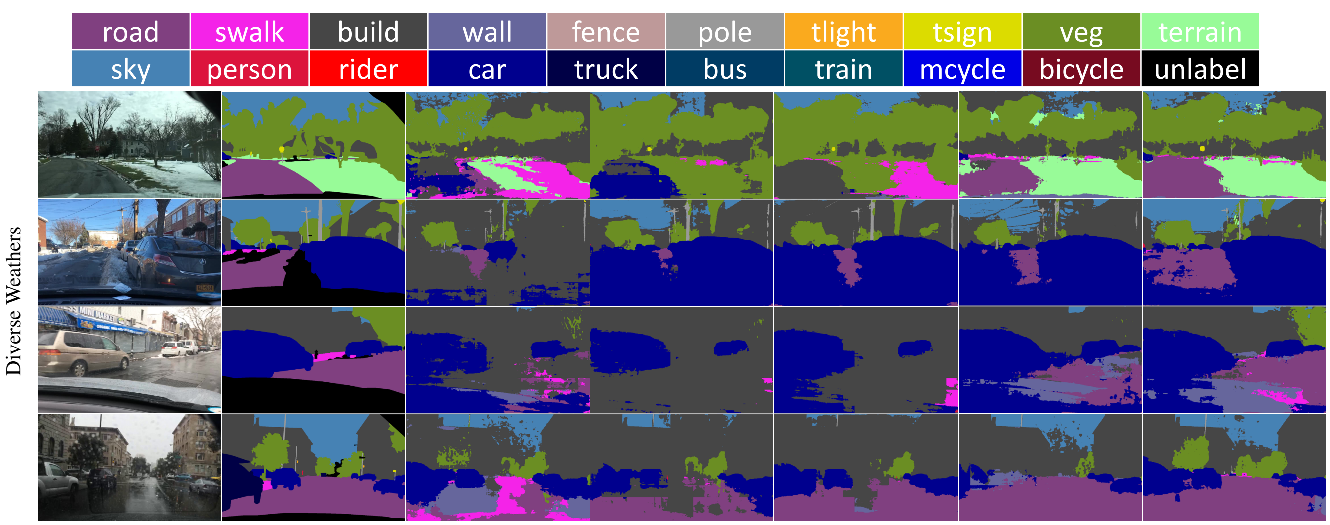
sunny), the BDD100K includes images acquired in various weather conditions, time zones, and locations. To compare the performance with regard to the variants of weather conditions, in Fig. 5, we selected the images taken in snowy or rainy weather conditions, and the baseline showed the vulnerable performance to this change. The normalization-based and vanilla meta-learning-based methods were also sensitive to visual changes in the road or sky caused by snow and rain. In contrast, our method predicted less damaged maps and showed reasonably estimation results on roads, sky, and vegetation regions. Fig. 6 shows the segmentation results under illumination and time changes. In addition, Fig. 6 shows the prediction maps under object visual changes due to the reflection of car glass, road slope, or unseen structures. To sum up, our method showed more robust results with respect to various visual changes existing in the real world than other DG methods.

Finally, Fig. 7 and Fig. 8 show the segmentation results on the Mapillary dataset. The Mapillary dataset contains images acquired from various environments in Europe and Asia. Our method showed more reasonable results than other methods even when the viewpoint or scene structure changes in places such as sidewalks, countryside, residential areas, and shoulder roads. Moreover, our method successfully predicted a partially snowy or wet road and cloudy sky. Therefore, we can describe that our memory-guided meta-learning method effectively enhances the encoder features on various distribution shifts.



(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [9] (e) RobustNet [1] (f) MLDG [10] (g) Ours

Figure 4. **Source (G+S)→Target (C):** Qualitative comparison on the Cityscapes dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)



(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [9] (e) RobustNet [1] (f) MLDG [10] (g) Ours

Figure 5. **Source (G+S)→Target (B):** [1/2] Qualitative comparison on the BDD100K dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)

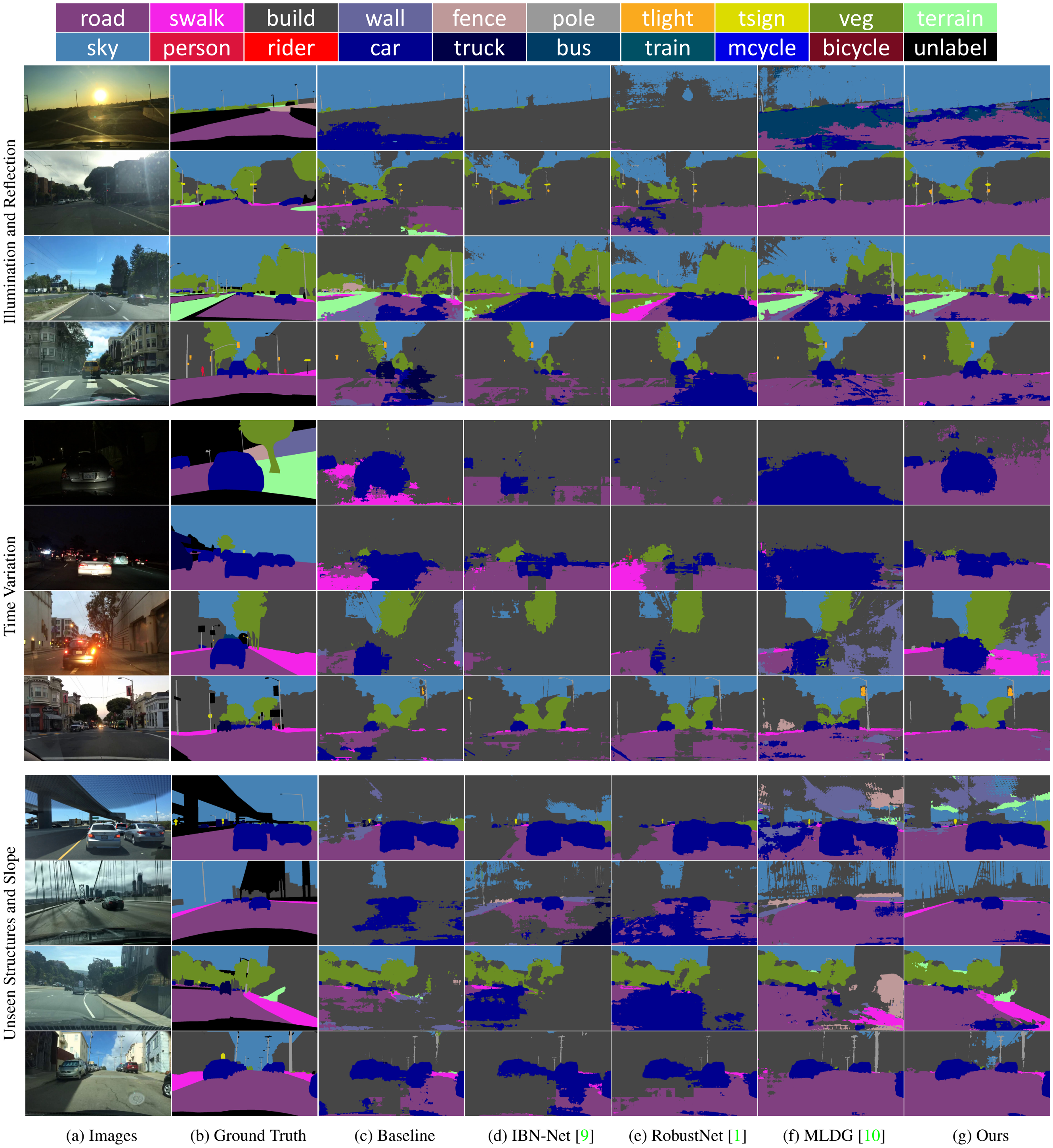


Figure 6. **Source (G+S)→Target (B):** [2/2] Qualitative comparison on the BDD100K dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)

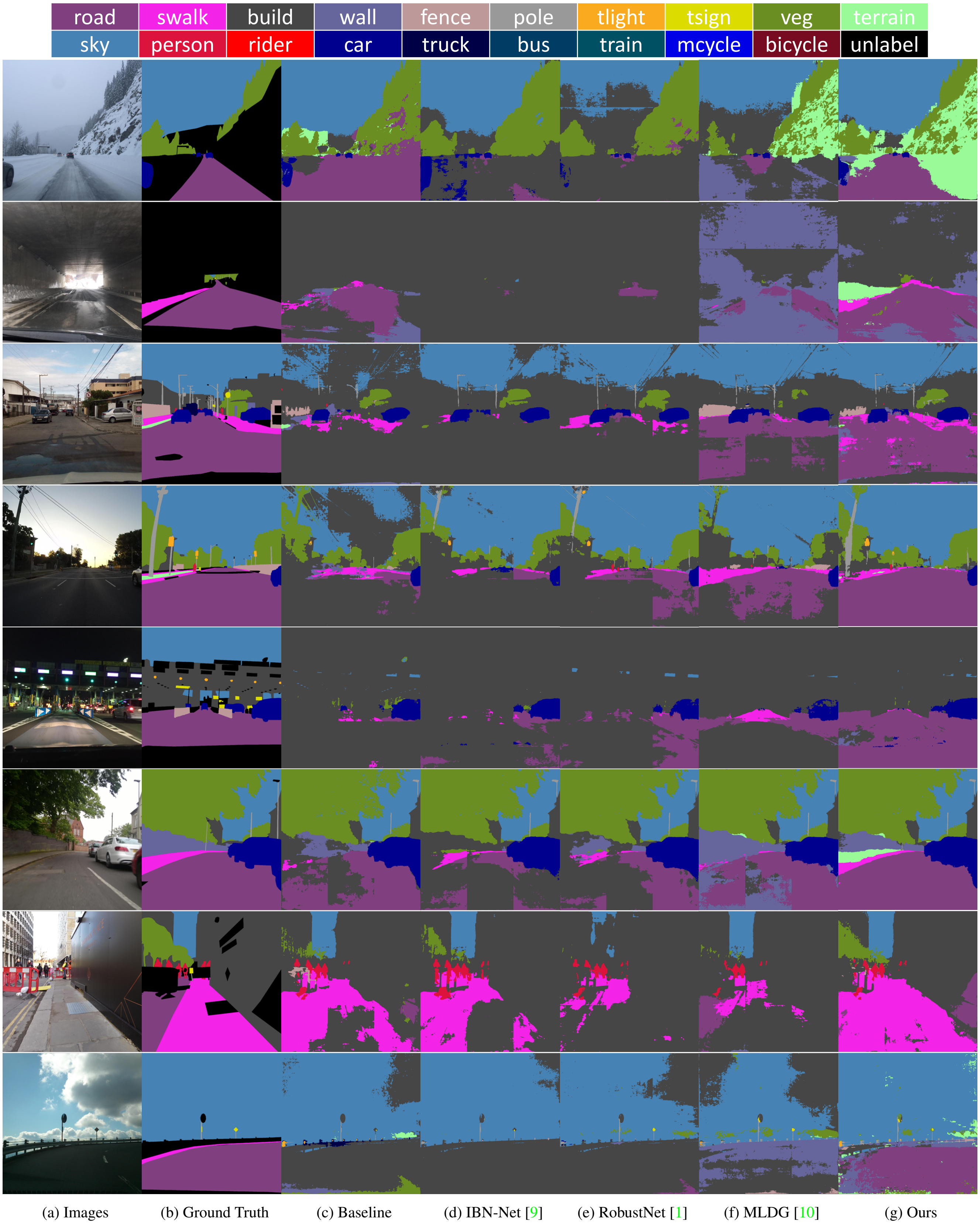
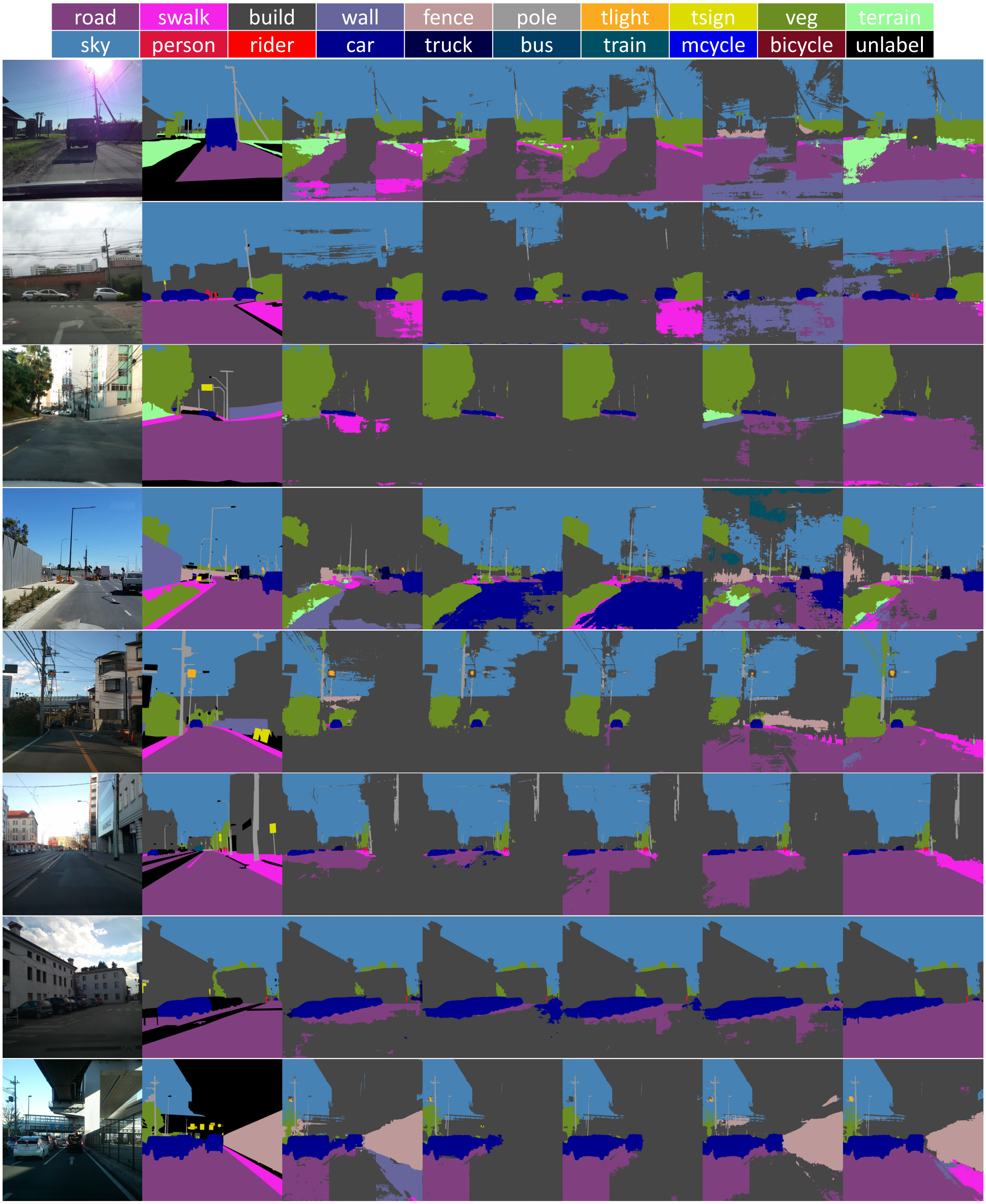


Figure 7. **Source (G+S)→Target (M):** [1/2] Qualitative comparison on the Mapillary dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)



(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [9] (e) RobustNet [1] (f) MLDG [10] (g) Ours

Figure 8. **Source (G+S)→Target (M):** [2/2] Qualitative comparison on the Mapillary dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)

References

[1] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, 2021. 1, 3, 4, 5, 6, 7, 8

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[4] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017. 1

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1

[8] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2018. 1

[9] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[10] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2, 3, 4, 5, 6, 7, 8

[11] Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Generalizable model-agnostic semantic segmentation via target-specific normalization. *PR*, 122:108292, 2022. 2, 4

[12] Zhenchao Jin, Tao Gong, Dongdong Yu, Qi Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *ICCV*, 2021. 3

[13] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, 2019. 3

[14] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *CVPR*, 2021. 3

[15] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 4

[16] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *NIPS*, 2018. 4

[17] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. In *NIPS*, 2019. 4

[18] Sicheng Zhao, Bo Li, Pengfei Xu, Xiangyu Yue, Guiguang Ding, and Kurt Keutzer. Madan: multi-source adversarial domain aggregation network for domain adaptation. *IJCV*, pages 1–26, 2021. 4

[19] Jianzhong He, Xu Jia, Shuaijun Chen, and Jianzhuang Liu. Multi-source domain adaptation with collaborative learning for semantic segmentation. In *CVPR*, 2021. 4

[20] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 3

[21] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 3

[22] 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 *CVPR*, 2016. 3

[23] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 3

[24] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 3