In this document, we describe second-order gradient flow of our method and details of experiments, and provide additional ablative 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 $M$ 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 $\beta$ was 1e-2 initially and decreased with exponential learning rate policy with the gamma of 0.9. The learning rate of the meta-training step $\alpha$ was set to 1/4 of the outer learning rate $\beta$ 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, $\lambda_1$ and $\lambda_2$, was set to 0.02 and 0.2, respectively.

#### 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].

---

*Corresponding authors.

1https://github.com/shachoic/RoustNet
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.

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

---

\(^2\)https://github.com/koncle/TSMLDG
still shows competitive performance against IBN-Net [9] and MLDG [10] due to the help of memory items. Moreover, \( L_{\text{coh}} \) and \( L_{\text{gph}} \) 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, 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 GTA5. 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., GTA5 [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...
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.

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 †.

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

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.)
Table 1. Comparison of different methods on the Mapillary dataset. The baseline is DeepLabV3+ with ResNet50. (Best viewed in color.)
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


[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


[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
