Pin the Memory: Learning to Generalize Semantic Segmentation

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

Abstract

The rise of deep neural networks has led to several breakthroughs for semantic segmentation. In spite of this, a model trained on source domain often fails to work properly in new challenging domains, that is directly concerned with the generalization capability of the model. In this paper, we present a novel memory-guided domain generalization method for semantic segmentation based on meta-learning framework. Especially, our method abstracts the conceptual knowledge of semantic classes into categorical memory which is constant beyond the domains. Upon the meta-learning concept, we repeatedly train memory-guided networks and simulate virtual test to 1) learn how to memorize a domain-agnostic and distinct information of classes and 2) offer an externally settled memory as a class-guidance to reduce the ambiguity of representation in the test data of arbitrary unseen domain. To this end, we also propose memory divergence and feature cohesion losses, which encourage to learn memory reading and update processes for category-aware domain generalization. Extensive experiments for semantic segmentation demonstrate the superior generalization capability of our method over state-of-the-art works on various benchmarks.[1]

1. Introduction

Semantic segmentation, assigning a semantic class label to each pixel, is a classical research topic for visual understanding in computer vision. The recent tremendous progress in semantic segmentation has been dominated by deep neural networks trained on large amounts of densly annotated datasets. Despite its success, models trained with a given dataset (*source*) do not generalize well in a new domain (*target*) that the models have not seen during training. Overcoming the domain shift issue caused by the different data distributions of two domains is crucial to deal with unexpected and unseen data, especially for replacing human tasks such as medical diagnosing or autonomous driving.

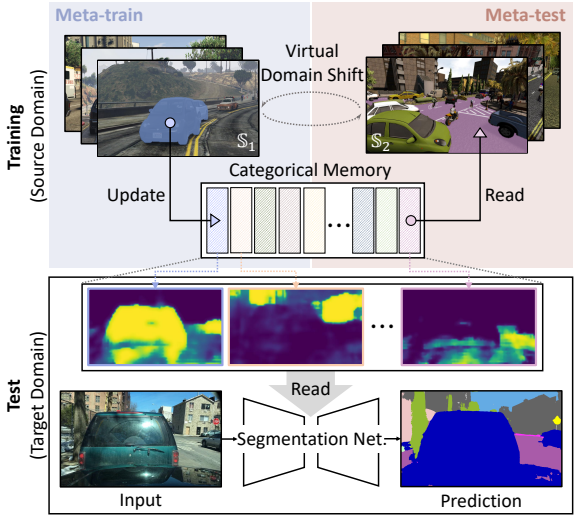


Figure 1. The illustration of our memory-guided meta-learning algorithm for domain generalization. Our method learns how to memorize domain-agnostic categorical knowledge that can provide an external guide to the test data in unseen target domain.

In order to mitigate severe performance degradation from the domain shift [5, 21], unsupervised domain adaptation (UDA) approaches [17, 42, 57] have been proposed to bridge the domain gap using unlabelled images of the target domain. These methods have introduced inventive learning strategies to learn domain invariant features [22, 29, 66, 67, 75, 78] or align source and target domain to unified space [23, 25, 52, 71, 72]. Though they have shown impressive results against domain shift, collecting data from the target domain is often impractical. Moreover, the scalability of the model is restricted as UDA requires network re-training or fine-tuning for the new target domain, thereby exposing limitations in terms of being able to generalize to 'any' unseen domains.

To overcome those limitations, domain generalization (DG) methods have been developed to learn robust models against variants of data distribution across arbitrary unseen domains [8, 30, 36, 38, 59, 79]. It is much harder than UDA in that no target domain data is available during training. Some

*Corresponding authors.

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF2021R1A2C2006703), the Yonsei University Research Fund of 2021 (2021-22-0001), and the Mid-Career Researcher Program through the NRF of Korea (NRF-2021R1A2C2011624).

[1]https://github.com/Genie-Kim/PintheMemory

methods heuristically define domain-biased information as style (*e.g*., texture, color) to explicitly augment it [28, 64], or erase style through instance normalization [47] and channel covariance whitening [14]. Despite their efforts, existing methods still show limited performance for use in real-world applications. But, it is natural that human visual system adapts stably even when facing scenes that they have never seen before. Where does this difference in generalization ability between humans and machines come from?

We argue that there is an important missing piece in this puzzle. The conceptual knowledge of humans [6], also known as semantic memory, is abstracted from actual experiences in the reusable form and is generalized to support a variety of cognitive activities such as event reconstruction [31, 32] and object recognition [54]. Inspired by this, we consider that human's knowledge concept can be effectively utilized in domain generalization by remembering the shared information of each class. For example, the style of the car may vary depending on the domain, but the basic features to configure the car (*e.g*. wheel, door, bumper, headlight) remain unchanged. Namely, the guidance of such prior knowledge about concurrent features can help to improve the generalization capability of machines.

In this work, we propose a novel memory-guided meta-learning framework to capture and memorize co-occurrent categorical knowledge between objects of the same class across domains. The objective of this framework is to assign shared information of each class into external memory slots and reuse the categorical concept for robust semantic segmentation in arbitrary unseen domains. To this end, we split source domain data into meta-training and meta-testing sets to explicitly mimic domain shift in the inference, allowing the network to store and invoke memory corresponding to domain-agnostic prototypes of class patterns, as shown in Fig. 1. That is, our method enables category-aware generalization for semantic segmentation, unlike previous DG approaches [14, 47, 64] that only concentrate on globally inferring domain-agnostic representations. Moreover, we introduce a memory divergence loss and a feature cohesion loss which boost discriminative power of memory and make more domain-invariant representations from the encoder, respectively. Consequently, our method achieves superior performance gain over existing DG approaches on multiple unseen real-world benchmarks. Without re-training or fine-tuning, our results are even on par with the multi-source UDA methods [23, 70–72], where the training images are given from both source and target domains.

In summary, our key contributions are as follows: (i) We present a novel approach to domain generalization for semantic segmentation with memory module to exploit domain-agnostic categorical knowledge of classes. (ii) We introduce the memory-guided meta-learning algorithm that improves the representation power of the memory-guided feature by exposing the model to mismatched data distribution. (iii) We propose two complementary losses, including memory divergence loss and feature cohesion loss, that promote power for an embedded feature to find the apposite class memory. (iv) Extensive experiments prove the significance of category-aware generalization on both single- or multi-source settings.

2. Related Work

Domain adaptation and generalization. There are wide investigations towards better generalization of deep networks to mitigate the domain distribution discrepancy between source (training) and target (testing) domains. In particular, unsupervised domain adaptation (UDA) approaches have been proposed to rectify such domain mismatch by leveraging the unlabeled target images for training [9, 17, 25, 35, 42, 57, 66, 68]. Recently, multi-source UDA methods [23, 70–72] have been introduced in a more practical scenario, where the training data is collected from multiple synthetic datasets [20, 49, 50]. Despite those efforts, deep networks often suffer from unseen novel domains in the real-world. It yields a domain generalization (DG) problem [7], that is more challenging than UDA in that the target domain data is not available. Recent works on DG are roughly categorized as two-fold: learning domain-invariant features [39, 43, 44, 79] and augmenting the training samples [8, 30, 59, 76, 77]. However, the majority of the DG methods still focuses on the task of classifying the entire image into one class, while our approach aims to generalize the networks to prevent a large performance drop of semantic segmentation in urban scene.

Domain generalization for semantic segmentation. This task has received relatively less attention compared to its importance in many real-world applications including autonomous driving in the wild. One of the promising solutions is a domain randomization [28, 64] to generate new training samples using data augmentation. However, it requires a lot of cost for training and it is practically difficult to cover the real-world distribution with the data augmentation only. Alternatively, based on the theoretical intuition from the normalization, some approaches have tried to normalize global feature by erasing the style-specific information of each domain [14, 47, 55]. In contrast to those methods that concern global representation only, we propose a categorical memory-guided framework for class-wise domain generalization. Meanwhile, recent papers [12, 13] pointed out a crucial role of the diversity of learned feature from the synthetic data to prevent overfitting to the source domain in segmentation task. Inspired by this, we employ a meta-learning to our framework for virtually testing the stored memory under different data distribution, promoting the only common knowledge of class to be saved for generalization.

Meta-learning. The model-agnostic meta-learning [18, 19] is one of the most popular methods of meta-learning (a.k.a

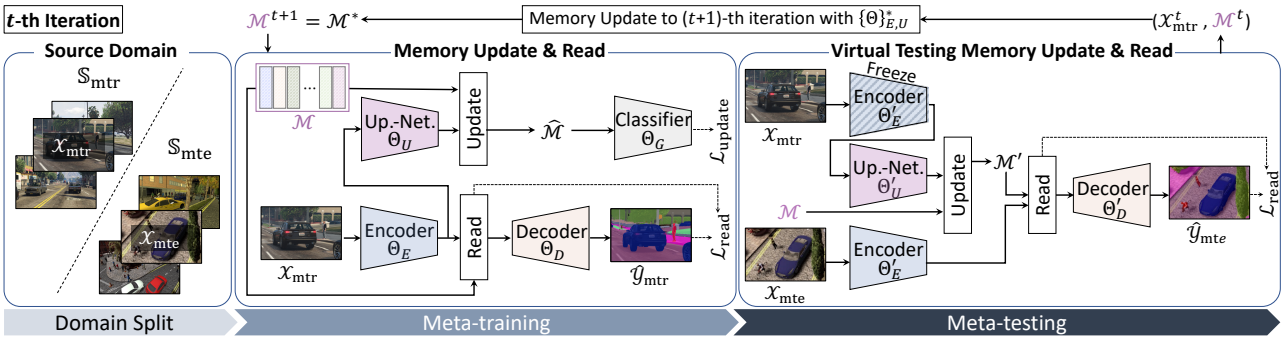


Figure 2. Overall training process of our method, consisting of domain split, meta-training and meta-testing steps for every iteration.

learning-to-learn), where an episodic training scheme has been designed for making multi-order of gradient descent for few-shot learning. The key idea of the episodic training, separating the learning steps into meta-train and meta-test to mimic the training and evaluation steps, has inspired other studies [3, 16, 34, 36–38, 40] to develop meta-learning based methods for domain generalization. Most related to ours, Zhen *et al*. [74] recently proposed a long-term memory with meta-learning that stores semantic information for few-shot learning, where the gradient from the updating memory does not feedback to the networks. Zhao *et al*. [73] claimed that the asynchronous gradient update among the sub-networks destabilizes meta-optimization, and simply treated the memory as a non-parametric module to solve the problem. Our method is orthogonal to these works in that we aims to learn the network to generalize categorical memory update and reading process through meta-learning.

Memory networks. The recent advances of memory networks [4, 48, 53] enhance the capability of neural networks by recording information stably. Although [61, 74] proposed the long-term memory modules with meta-learning like our method, they improved reading performance only without consideration for memory writing. Compared to the previous works, our memory module stores long-term memory in whole training steps with meta-learning, which helps to robustly read and write memory to domain shift. The memory in [4] approximated to neural networks requiring several computations to read memory, but our method is more efficient than [4] with a once estimation. Significantly, the memory networks have been effective in several segmentation-related tasks [1, 26, 27, 33, 46, 60–62]. For instance, Jin *et al*. [33] stored dataset-level surrounding contexts of various classes to augment pixel-level representations. On the contrary, we store domain-agnostic information into the memory to contain common features of semantic categories.

3. Proposed Method

3.1. Problem Statement and Overview

Given an image from an unseen target domain, domain generalization aims to protect the performance of the segmentation network trained with a set of observable source, $\mathbb{S}$, where the networks basically consist of encoder and decoder (pixel-wise classifier). An intuitive approach to DG is to learn the segmentation networks by simply combining all source domains into one training dataset and training with standard segmentation loss such as cross-entropy [41]. However, this naive aggregation method is overly suited to the source domain and thus shows enormous performance deterioration when domain shift occurs in the inference.

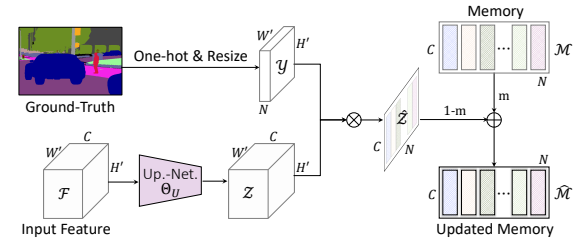
To solve this problem, we propose a memory-guided meta-learning framework to prevent performance degradation of semantic segmentation in the unseen domains at test time, as shown in Fig. 2. By configuring an artificial domain shift with data augmentation or domain splitting, we allow the network to update and read memory on the specified domains in the meta-learning framework so that the network learns how to remember conceptual knowledge in the presence of the domain shift. In the following section, we first describe the memory read and update procedure (Sec. 3.2) and then memory-guided meta-learning framework with loss functions (Sec. 3.3).

3.2. Memory Module

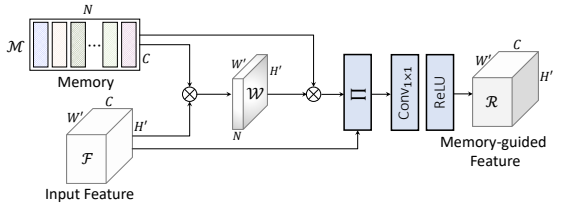
The memory module is incorporated with the segmentation backbone network to memorize the common features of each class into memory matrix $\mathcal{M} \in \mathbb{R}^{N \times C}$, where N is the number of classes and C is a channel dimension of the encoder feature. We next explain initialization, update and reading processes of our memory module in details.

Initialization. As the preliminary step, ℓ_2-normalized feature maps are extracted from all training images in the source domains through an encoder E with parameters Θ_E, pre-trained on ImageNet [51]. To initialize the memory matrix with these feature maps, we calculate a mean feature vector for each class by masking the regions with ground-truth segmentation maps. Since the initial memory matrix composing of class-wise mean vectors is in a very noisy state, our method learns to update this by storing more discriminative and domain-agnostic class-wise features in the memory.

Update. We adopt a memory updating network U consists of a 1×1 convolution layer with the residual connection.



(a) Memory Update



(b) Memory Read

Figure 3. Illustration of memory update and reading operations.

As shown in Fig. 3a, the memory updating network with parameters Θ_U transforms ℓ_2-normalized feature map $\mathcal{F} \in \mathbb{R}^{C\times H'\times W'}$ of an input image $\mathcal{X} \in \mathbb{R}^{3\times H\times W}$ into $\mathcal{Z} = U(\mathcal{F})$, where $H \times W$ is an original size of the image, and $H' \times W'$ is a reduced size by a pooling operation in the backbone networks[2]. In order to update the n-th item $\mathcal{M}[n]$ in the class-wise memory, we perform an average pooling over the masked region by referring to the segmentation mask of the n-th class as follows:

$$\hat{\mathcal{Z}}[n] = (\mathcal{Y}[n]\mathcal{Z}^\top)/K_n, \tag{1}$$

where K_n is the number of pixels belonging to n-th class in the ground-truth, $\hat{\mathcal{Z}} \in \mathbb{R}^{N\times C}$ is a masked feature map and $\mathcal{Y}$ is a one-hot segmentation ground-truth which has a size of $N \times H'W'$. Note that $\mathcal{Z}$ is reshaped as $C \times H'W'$. Then the n-th channel of masked feature vector $\hat{\mathcal{Z}}[n]$ is used to update a memory item using moving average.

$$\hat{\mathcal{M}}[n] = m \cdot \mathcal{M}[n] + (1-m)\cdot \hat{\mathcal{Z}}[n], \tag{2}$$

where $\hat{\mathcal{M}}[n]$ is an updated memory and the momentum m is set as 0.8 empirically. This is repeated for all classes, which is expressed as below:

$$\hat{\mathcal{M}} = \textbf{update}(\mathcal{M}, \mathcal{X}; \{\Theta\}_{E,U}), \tag{3}$$

where a set of parameters Θ_E and Θ_U is denoted as $\{\Theta\}_{E,U}$.

Read. As depicted in Fig. 3b, we read the stored memory items with the encoded feature map $\mathcal{F}$ to represent a memory-guided feature map $\mathcal{R} \in \mathbb{R}^{C\times H'\times W'}$ which is used in the decoder. To aggregate a corresponding memory item along each feature location, we compute an memory weight matrix $\mathcal{W} \in \mathbb{R}^{N\times H'\times W'}$ via cosine similarity and normalize it with softmax function as:

$$\mathcal{W}[n] = \frac{\exp(\mathcal{M}[n]\mathcal{F})}{\sum_{n'=1}^{N}\exp(\mathcal{M}[n']\mathcal{F})}, \tag{4}$$

where $\mathcal{F}$ and $\mathcal{W}$ are permuted as $C \times H'W'$ and $N \times H'W'$ respectively. The memory-guided feature map $\mathcal{R}$ is obtained by fusing the original feature map $\mathcal{F}$ and weighted memory feature $\mathcal{M}^\top\mathcal{W}$ as follows:

$$\mathcal{R} = \texttt{ReLU}(\texttt{Conv}_{1\times1}(\Pi(\mathcal{F}, \mathcal{M}^\top\mathcal{W}))), \tag{5}$$

where $\Pi(\cdot)$ denotes a concatenation operation. Note that $\mathcal{M}^\top\mathcal{W}$ is re-permuted to have a size of $C \times H' \times W'$. We add 1×1 convolution layer to make the channel size of $\mathcal{R}$ to C. Finally, a predicted segmentation probability map $\hat{\mathcal{Y}}$ is estimated by passing $\mathcal{R}$ into the decoder. From now on, we denote the 1×1 convolution layer with decoder as D with parameters Θ_D.

3.3. Learning to Generalize Update and Read

Compared to the previous DG methods based on meta-learning [3, 36, 37] that do not use external prior knowledge, our method leverages meta-learning to achieve two goals. First, the domain invariant categorical knowledge of each class is saved in a form of external memory that can offer a class-wise guidance for robustly segmenting an image from unseen domains. Second, we reinforce our network to robustly classify each unseen image pixel to a category label against intra-class and cross-domain variations. Specifically, we randomly split the available source domains $\mathbb{S}$ into meta-train domains $\mathbb{S}_{\text{mtr}}$ and meta-test domains $\mathbb{S}_{\text{mte}}$ at every iteration step. Then, we repeatably memorize class-wise features from $\mathbb{S}_{\text{mtr}}$ and test whether the network properly works with the memory on the held-out $\mathbb{S}_{\text{mte}}$. The overall training procedure is summarized in Fig. 2 and Alg. 1.

Meta-training. Given an input image $\mathcal{X}_{\text{mtr}} \in \mathbb{S}_{\text{mtr}}$, the encoder computes a feature map $\mathcal{F}_{\text{mtr}}$ and augments it by using the memory $\mathcal{M}$ through the reading operation. We calculate a per-pixel cross-entropy loss [41], *i.e.* segmentation loss $\mathcal{L}_{\text{seg}}$, with ground-truth map $\mathcal{Y}_{\text{mtr}}$ and the estimated output $\hat{\mathcal{Y}}_{\text{mtr}}$ from the decoder. However, $\mathcal{L}_{\text{seg}}$ does not necessarily guarantee that the encoder features in the same class lie close in the feature embedding space. Therefore, we further propose a feature cohesion loss $\mathcal{L}_{\text{coh}}$ to encourage semantic features to be locally assembled based on each memory item:

$$\mathcal{L}_{\text{coh}} = \frac{1}{H'W'}\sum_{j=1}^{H'W'} -\mathcal{Y}_{\text{mtr}}^\top[j]\log(\mathcal{W}_{\text{mtr}}[j]), \tag{6}$$

where $\mathcal{W}_{\text{mtr}}$ is computed as (4).

In addition, the class-wise features in the memory should be far enough apart from each other to be discriminative. To ensure this, we propose a memory divergence loss $\mathcal{L}_{\text{div}}$ that

[2] H', W' varies depending on the output stride of backbone networks such as FCN [41], DeepLabV2 [10], DeepLabV3+ [11], etc.

increases the distance between memory items, as well as maximizes the decision margin:

$$\mathcal{L}_{\text{div}} = \sum_{n=1}^{N}(-\mathcal{I}[n]\log(G(\hat{\mathcal{M}}[n]^{\top})) + 2\cdot\sum_{n'\neq n}^{N}\frac{\max(\hat{\mathcal{M}}[n]\hat{\mathcal{M}}[n']^{\top},0)}{N(N-1)}), \tag{7}$$

where $\mathcal{I}$ is the identity matrix of size $N\times N$, and a memory classifier G includes a FC layer with parameters Θ_G and has an output size of N after the softmax. In (7), the first term is for the memory classification, and the second term is similar to cosine embedding loss [58] with a margin set to 0, empirically scaled double. While the divergence loss improves inter-class dispersion, the feature cohesion loss increases intra-class compactness of the encoder features among distinct memory items. We carefully note that $\mathcal{L}_{\text{div}}$ is calculated for the newly estimated memory $\hat{\mathcal{M}}$, while the reading process use the $\mathcal{M}$ updated in the last iteration step. It is because the reading process aims to guide the feature map well with previously saved memory, and the update process focuses on saving even better patterns into memory and widening the gap between memory items with $\mathcal{L}_{\text{div}}$.

To clarify, we define $\mathcal{L}_{\text{read}}$ with the segmentation and feature cohesion losses computed in the memory reading operation, and $\mathcal{L}_{\text{update}}$ with the memory divergence loss computed when updating the memory item, respectively:

$$\begin{aligned}\mathcal{L}_{\text{read}}(\mathcal{M},\mathcal{X}_{\text{mtr}};\{\Theta\}_{E,D}) &= \mathcal{L}_{\text{seg}}+\lambda_1\mathcal{L}_{\text{coh}},\\ \mathcal{L}_{\text{update}}(\mathcal{M},\mathcal{X}_{\text{mtr}};\{\Theta\}_{E,U,G}) &= \lambda_2\mathcal{L}_{\text{div}},\end{aligned} \tag{8}$$

where λ_1 and λ_2 are hyper-parameters. Consequently, the updated network parameters are obtained as follows:

$$\begin{aligned}\{\Theta\}'_{E,U,D},\Theta^*_G \leftarrow \{\Theta\}_{E,U,D,G} &\\ -\alpha\nabla_{\Theta}\mathcal{L}_{\text{read}}(\mathcal{M},\mathcal{X}_{\text{mtr}};\{\Theta\}_{E,D}) &\\ -\alpha\nabla_{\Theta}\mathcal{L}_{\text{update}}(\mathcal{M},\mathcal{X}_{\text{mtr}};\{\Theta\}_{E,U,G}),&\end{aligned} \tag{9}$$

where α is a learning rate of the meta-training step. Since the memory classifier G is not used in the meta-testing step, Θ^*_G is the final updated parameter of G in this training iteration.

Meta-testing. The goal of meta-testing in our method is to not only virtually simulate *testing the networks* on new data statistics but also characterize *learning to update categorical memory* to work well across the domains. Moreover, the effectiveness of the memory divergence loss for the updating network U should be tested within the meta-testing process.

With these reasons, we carefully design meta-testing process that re-updates the memory using the meta-updated networks' parameters $\{\Theta\}'_{E,U}$ and the meta-train image $\mathcal{X}_{\text{mtr}}$:

$$\mathcal{M}' = \textbf{update}(\mathcal{M},\mathcal{X}_{\text{mtr}};\text{copy}(\Theta'_E),\Theta'_U), \tag{10}$$

where $\text{copy}(\Theta'_E)$ indicates Θ'_E is frozen. We obtain the memory once again with meta-train data, not meta-test data, because we will reuse the learned memory without update process in inference. Since this memory $\mathcal{M}'$ is used to segment meta-test data $\mathcal{X}_{\text{mte}}$, this novel step also allows the memory updating network's parameters Θ_U to receive the second-order gradient feedback on whether the updated memory $\mathcal{M}'$ is applicable on different domains. By freezing the encoder's parameter Θ'_E, we can avoid unstable meta-learning caused by the asynchronous gradient update between the encoder and the other networks [73]. Guided by $\mathcal{M}'$, the network parameters are updated with the reading loss $\mathcal{L}_{\text{read}}$ for the image $\mathcal{X}_{\text{mte}}$ from meta-test domain $\mathbb{S}_{\text{mte}}$ as follows:

$$\begin{aligned}\{\Theta\}^*_{E,U,D} \leftarrow &\{\Theta\}_{E,U,D}\\ &-\beta\nabla_{\Theta}\mathcal{L}_{\text{read}}(\mathcal{M}',\mathcal{X}_{\text{mte}};\{\Theta\}'_{E,U,D}),\end{aligned} \tag{11}$$

where β is a learning rate of the meta-testing step. Note that the second-order gradient is generated from the last term of (11) by differentiating $\{\Theta\}'$ obtained from (9) with the original parameters $\{\Theta\}$. Using the updated network parameters, we initialize the memory $\mathcal{M}^*$ that will be used in the next training iteration step:

$$\mathcal{M}^* = \textbf{update}(\mathcal{M},\mathcal{X}_{\text{mtr}};\text{copy}(\{\Theta\}^*_{E,U})). \tag{12}$$

The optimization in meta-testing step allows (1) writing the domain-agnostic features to the current memory $\mathcal{M}$ from the meta-train image in (12) and (2) ensuring the generalization capability of the memory-guided feature of meta-test image.

Algorithm 1: Overall Training Procedure

```
Initialize {Θ}_{E,U,D,G} and M at t = 0
while t < T do
    Randomly split S into S_mtr and S_mte
    Meta-training:
        Sample batch X^t_mtr = {X^b_mtr}_{b=1}^B from S_mtr
        Compute L_read with (X^t_mtr, M, {Θ}_{E,D})
        M̂ ← update(M, X^t_mtr; {Θ}_{E,U})
        Compute L_update with (M̂, Θ_G)
        Update {Θ}'_{E,U,D}, Θ*_G from {Θ}_{E,U,D,G} in (9)
    Meta-testing:
        M' ← update(M, X^t_mtr; copy(Θ'_E), Θ'_U)
        Sample batch X^t_mte = {X^b_mte}_{b=1}^B from S_mte
        Compute L_read with (X^t_mte, M', {Θ}'_{E,D})
        Update {Θ}*_{E,U,D} from {Θ}'_{E,U,D} in (11)
        M* ← update(M, X^t_mtr; copy({Θ}*_{E,U}))
    {Θ}_{E,U,D,G} ← {Θ}*_{E,U,D,G}
    M ← M*
    t ← t + 1
```

4. Experiments

4.1. Experimental Setup

Datasets. We conduct the experiments on six different datasets to prove the generalization ability of our method.

- **Real datasets:** Cityscapes [15] includes 3,450 finely-annotated images collected from 50 different cities, primarily Germany. We use only a finely-annotated set for

	Methods	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	m-bike	bicycle	mIoU(%)
Cityscapes	Baseline†	72.7	36.4	64.9	11.9	2.8	31.0	37.7	20.0	84.9	14.0	71.9	65.3	9.9	84.7	11.6	25.4	0.0	10.6	18.1	35.46
	IBN-Net† [47]	68.3	29.5	69.7	17.4	1.8	30.7	36.2	20.2	85.4	18.2	81.8	64.7	12.9	82.7	13.0	16.2	0.0	8.2	22.2	35.55 (0.1)
	RobustNet† [14]	82.6	40.1	73.4	17.4	1.4	34.2	38.6	18.5	84.9	16.9	81.9	65.2	11.4	84.7	7.2	23.6	0.0	10.4	**23.9**	37.69 (2.2)
	Baseline	49.1	28.0	69.8	21.1	12.2	21.5	**39.3**	13.0	81.8	**33.7**	68.7	66.0	18.2	38.1	20.7	15.6	3.6	16.4	18.4	33.42
	MLDG‡ [36]	75.8	37.4	78.1	**27.6**	8.5	**37.4**	31.6	18.7	84.0	16.2	70.2	**66.3**	16.7	74.0	20.4	**38.4**	0.0	20.4	16.1	38.84 (5.4)
	Ours	**85.3**	**45.3**	**82.5**	26.3	**19.9**	34.9	39.0	**24.0**	**85.8**	24.0	**82.8**	64.7	**21.3**	**85.7**	**32.0**	38.2	**6.7**	**26.0**	21.5	**44.51 (11.1)**
BDD100K	Baseline†	44.6	26.1	34.7	1.8	6.9	29.5	39.1	20.5	64.9	10.8	51.6	50.6	10.2	63.9	1.1	4.8	0.0	5.5	10.1	25.09
	IBN-Net† [47]	53.8	25.0	55.4	2.8	14.8	32.9	39.7	26.3	**71.7**	16.4	85.9	**57.4**	**17.5**	56.9	5.3	6.0	0.0	18.5	**25.4**	32.18 (7.1)
	RobustNet† [14]	69.5	35.0	60.9	4.1	13.1	**36.6**	**40.5**	**27.3**	71.6	14.0	83.6	56.0	17.3	61.9	4.4	8.8	0.0	24.3	18.9	34.09 (9.0)
	Baseline	54.5	26.0	44.0	3.4	20.9	30.1	37.4	15.9	65.7	22.7	42.3	50.9	14.7	58.0	17.5	14.1	0.0	**25.0**	9.4	29.07
	MLDG‡ [36]	54.0	33.4	61.0	**6.4**	25.3	35.5	35.5	19.0	71.5	20.0	75.8	53.7	13.4	46.2	7.3	**34.4**	0.0	9.5	5.3	31.95 (2.9)
	Ours	**79.3**	**39.1**	**69.0**	6.2	**32.8**	32.1	36.7	26.9	71.3	**25.9**	**86.3**	49.4	12.5	**75.2**	**20.6**	31.6	0.0	17.9	10.7	**38.07 (9.0)**
Mapillary	Baseline†	62.0	36.3	32.5	9.5	7.7	29.9	40.5	22.5	78.6	**40.9**	61.0	59.4	6.4	78.3	5.1	5.1	0.1	9.0	21.8	31.94
	IBN-Net† [47]	67.4	38.8	51.3	10.2	7.6	36.0	40.1	40.8	**80.3**	39.9	**92.1**	61.8	14.0	74.4	10.7	9.4	3.5	15.3	**25.4**	38.09 (6.2)
	RobustNet† [14]	78.0	**41.0**	56.6	13.1	6.2	**39.4**	**41.3**	36.1	79.5	34.7	90.0	61.0	12.0	76.1	10.7	13.1	0.8	16.9	24.8	38.49 (6.6)
	Baseline	53.4	25.9	44.7	11.1	19.0	28.4	36.2	15.8	71.3	27.1	66.1	58.6	11.7	64.2	20.1	1.1	**11.4**	**23.1**	22.3	32.19
	MLDG‡ [36]	69.4	36.0	58.6	**19.4**	16.8	37.6	31.3	28.8	76.7	36.9	81.6	43.4	15.5	59.1	21.4	8.1	1.3	16.8	17.9	35.60 (3.7)
	Ours	**78.0**	40.8	**71.1**	14.6	**27.0**	34.2	40.7	**50.3**	77.1	26.2	90.0	**63.1**	**24.0**	**81.6**	**30.5**	**15.5**	5.3	18.7	22.7	**42.70 (10.5)**

Table 1. **Source (G+S)→Target (C, B, M):** Mean IoU(%) and per-class IoU(%) comparison of other SOTA DG methods for semantic segmentation. We report the mIoU improvement as red text. The networks are DeepLabV3+ with ResNet50 and results with † are from [14].

training and validation. **B**DD100K [63] contains 8K diverse urban driving scene images collected from various locations in the US. **M**apillary [45] is a real street-view dataset including 25K images collected from all around the world. **I**DD [56] contains 10,004 images captured from Indian roads. The road scenes in the IDD, which contain animals and muddy, are significantly different from the existing datasets mainly collected in Europe or US.

- **Synthetic datasets: G**TAV [49] includes 24,966 driving-scene images generated from a game engine. It has 19 object categories compatible with the real-world datasets. **S**ynthia [50] is another synthetic dataset simulating different seasons, weather, and illumination conditions from multiple viewpoints. The Synthia dataset contains 9,400 photo-realistic synthetic images annotated into 16 categories compatible with the GTAV.

Metrics. Following the standard-setting [14, 23], we report mean Intersection over Union (mIoU) averaged over all classes to measure the segmentation performance.

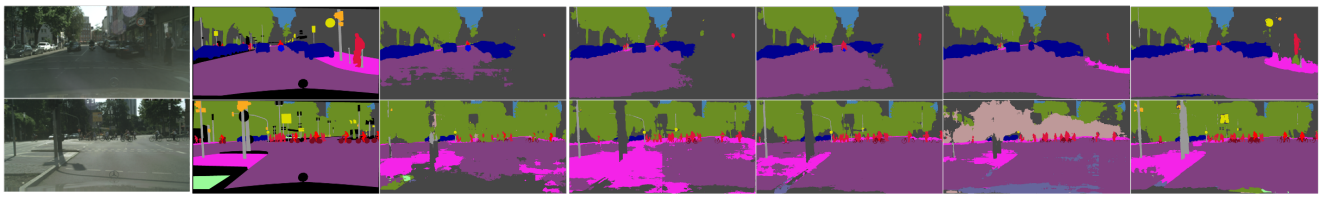
Implementation details. We conducted experiments by adopting DeepLabV3+ [11] and DeepLabV2 [10] with ResNet50 and ResNet101 [24] as a semantic segmentation architecture, respectively, where the output stride is 16 for DeepLabV3+. All backbones were initialized with ImageNet [51] pre-trained model. We set the maximum iterations to 120K but early stop at 30K iterations, except for ResNet-101 models trained for 70K. The hyper-parameters, λ_1 and λ_2, were empirically set to 0.02 and 0.2. Further details for optimization and training are explained in the supplementary material. In all experiments, we denote the networks trained with aggregated source domains as a *baseline*. To conduct experiments, we re-implemented several DG methods and marked them with ‡.

Methods	Cityscapes	BDD100K	Mapillary	Avg.
Baseline	52.51	47.47	54.70	51.56
IBN-Net‡ [47]	54.39	48.91	56.06	53.12
RobustNet‡ [14]	54.70	49.00	56.90	53.53
MLDG‡ [36]	54.76	48.52	55.94	53.07
TSMLDG‡ [65]	53.02	46.43	52.76	50.70
Ours	**56.57**	**50.18**	**58.31**	**55.02**

Table 2. **Source (G+S+I)→Target (C, B, M):** Mean IoU(%) comparison of other state-of-the-art DG methods, where all networks are trained with two synthetic (GTAV, Synthia) and one real (IDD) datasets. All methods adopt DeepLabV3+ with ResNet50.

4.2. Results

Comparison with state-of-the-art. Table 1 summarizes the test results on the most popular real-world dataset benchmarks, where the models were trained on multi-source domains (GTAV and Synthia). We conduct comparisons with the re-implemented vanilla meta-learning method without the memory module (MLDG) and the normalization-based methods (IBN-Net and RobustNet) based on the results reported in the paper [14]. While the existing normalization-based methods are slightly better than the baseline performance, our approach consistently outperforms the state-of-the-arts (SOTA) by a large margin on all real-world datasets. It demonstrates that the generalization methods by erasing the visual style of domains makes it hard to leverage multi-source domain information well. Especially, our approach shows significantly improved gain over baseline as 11.1% on the Cityscapes, 9.0% on the BDD100K and 10.5% on the Mapillary. Furthermore, compared to MLDG [36] that uses meta-learning framework like ours, our method proves the effectiveness of the categorical memory to improve the gen-



(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [47] (e) RobustNet [14] (f) MLDG [36] (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.)

	Methods	Cityscapes		Methods	Cityscapes
FCN-8s	Baseline	32.5	DeepLabV3+	Baseline†	29.0
	DRPC [64]	37.4		IBN-Net† [47]	33.9
	Baseline	21.4		RobustNet† [14]	36.6
	CNSN [55]	36.5		Baseline	31.6
	Baseline	23.3		MLDG [36]	36.7
	ASG [13]	31.9		**Ours**	**41.0**

Table 3. **Source (G)→Target (C):** Mean IoU(%) comparison of other SOTA methods with various segmentation models with ResNet50. Results with † are from [14]. The results on other datasets are reported in supplementary materials.

Methods	w/Target	Cityscapes	BDD100K
Baseline	✗	40.0	37.4
CyCADA [25]†	✓	39.3	37.2
MDAN [70]†	✓	36.0	29.4
MADAN [72]†	✓	45.4	40.4
MADAN+ [71]	✓	48.5	42.7
CLSS [23]	✓	**54.0**	N/A
Ours	✗	49.4	**45.5**

Table 4. **Source (G+S)→Target (C, B):** Mean IoU(%) comparison of other multi-source UDA methods. The segmentation models are all DeepLabV2 with ResNet101. Results with † are from [71].

eralization capability. Fig. 4 shows qualitative results and more results are provided in supplementary materials. To further verify the performance variation when more source data is used, we add one more real dataset (IDD) to the source domain following [65]. Since the IDD dataset significantly differs from the existing real datasets, this scenario assumes a new generalization scenario where the available real dataset looks very different from the target domain's culture. In Table 2, we can see that our method also outperforms all previous approaches in this setting.

Table 3 shows the results evaluated on the Cityscapes dataset with various segmentation models regarding to single-source domain generalization setting. Like [14], we generate virtual domain shift by photometric transformations such as Gaussian blur or color jitter in this setting. Even though the networks are trained on the GTAV dataset only, our method obtains the best generalization performance with a relatively high-performance gain. It thus points out that category-aware generalization, like our method, should be encouraged importantly to further research in this area.

Comparison with UDA. We also compared our result with state-of-the-art UDA methods [23, 70–72] trained on multi-

$\mathcal{L}_{\text{coh}}$	$\mathcal{L}_{\text{div}}$	Cityscapes	BDD100K	Mapillary	Avg.
✓		43.85	38.01	41.66	41.17
	✓	42.08	36.80	40.83	39.90
✓	✓	**44.51**	**38.07**	**42.70**	**41.76**

Table 5. Ablation study on the variants of loss.

Memory	Freeze E	Freeze U	Cityscapes	BDD100K	Mapillary	Avg.
$\hat{\mathcal{M}}$	-	-	40.64	31.06	31.59	34.43
$\mathcal{M}'$	✗	✗	41.65	36.56	38.80	39.00
$\mathcal{M}'$	✓	✗	**44.51**	**38.07**	**42.70**	**41.76**
$\mathcal{M}'$	✓	✓	41.67	32.04	33.90	35.87

Table 6. Ablation study on the variants of the memory update strategy in meta-testing step.

Training	Memory	M.L.	Cityscapes	BDD100K	Mapillary	Avg.
Agg.	✗	✗	33.42	29.07	31.90	31.46
Agg.	✓	✗	38.28	31.46	32.25	34.00
Episodic	✗	✓	38.84	31.95	35.60	35.46
Episodic	✓	✗	41.50	38.00	40.22	39.91
Episodic	✓	✓	**44.51**	**38.07**	**42.70**	**41.76**

Table 7. Ablation study on the variants of the memory learning method. We denote the baseline as 'Agg.', episodic training as 'Episodic' and the second-order gradient as 'M.L.'.

ple synthetic datasets. For a fair comparison, we reported mIoU score over 16 object classes in Table 4. All other methods excluding baseline and ours used training images from target domain to learn their models. It is interesting that even though UDA is a much easier setting than domain generalization, our DG method achieved the highest performance on the BDD100K. Except for CLSS [23], our method also showed competitive results on the Cityscapes. We argue that our method, guided by domain-agnostic categorical memory learned from multi-source domains, is more effective to deal with diverse real-world cases than UDA methods.

4.3. Ablation Study and Discussion

All experiments in this subsection uses the model trained on GTAV and Synthia datasets adopting DeepLabV3+ [11] with ResNet50 [24].

Loss. To verify the effectiveness of the proposed losses, we study different loss combinations of $\mathcal{L}_{\text{coh}}$ and $\mathcal{L}_{\text{div}}$. From Table 5, we observe that both loss terms make substantial contributions to the performance gain for all datasets by operating complementary each other.

Memory update strategy. In (10) of the meta-testing step, we freeze the encoder parameters and re-update the memory. To show the effectiveness of this update scheme, we con-

Methods	# of Params	GFLOPs	Time (ms)
Baseline	45.08M	277.16	13.51
IBN-Net‡ [47]	45.08M	277.24	14.31
RobustNet‡ [14]	45.08M	277.20	14.53
MLDG‡ [36]	45.08M	277.16	13.85
Ours	45.22M	277.69	13.56

Table 8. Comparison of computational cost. Tested with the image size of 2048×1024 on NVIDIA TITAN RTX GPU. We averaged the inference time over 500 trials.

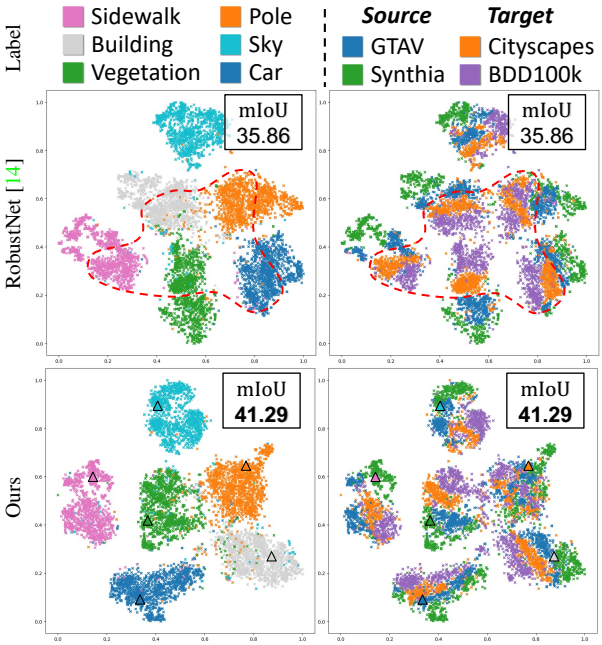


Figure 5. **Source (G+S)→Target (C, B):** t-SNE visualization of extracted features. Colors indicate different categories in the first column and different domains in the second column. Memory is pointed as triangle. The mIoU scores are the average scores of the target domains.

duct an ablation study with the combination for the encoder E and memory updating network U in Table 6. Without re-updating the memory in the meta-testing step, a severe generalization performance drop occurs since the updating network is not updated. Additionally, a similar degradation occurs when both encoder and the updating networks are frozen, because the generalization objective for memory updating is not evenly set. This means that our novel step is operated for memorizing domain-agnostic features while stabilizing meta-optimization.

Memory learning framework. We analyze our method by dividing it into three key contributing factors: memory, second-order gradient ('M.L.' [18] in Table 7) and episodic learning without 'M.L.' that creates an artificial domain shift environment. In Table 7, for the memory with episodic learning, we perform memory update on $\mathbb{S}_{\text{mtr}}$ and memory reading on $\mathbb{S}_{\text{mte}}$. In this process, we do not calculate the second-order gradient as no meta-testing step. To sum up, our method enhances the generalization capability most effectively with

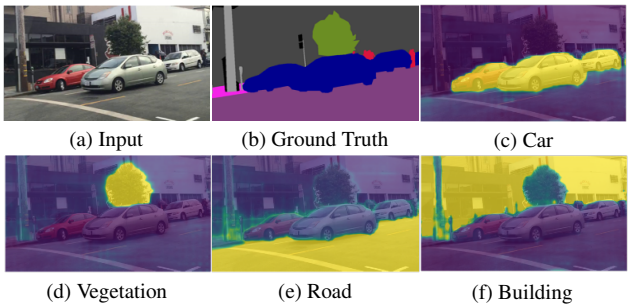


Figure 6. **Source (G+S)→Target (B):** Visualization of the channels of memory weight matrix from (4) with the BDD100K dataset.

the combination of all these variants.

Visualization of t-SNE. In Fig. 5, we visualize the pixel representations with the models learned with our method and normalization based DG method [14]. In the second column, we can see that the unseen domain features of RobustNet tend to agglomerate with each other. In contrast, our method significantly reduces the tendency that features belonging to the same domain but of different classes aggregate with each other, especially in pole class. At the same time, our method shows superior generalization performance than RobustNet. It demonstrates that our method effectively integrates source domain information by generalized categorical knowledge.

Running time and complexity. In Table 8 we compare with exisiting DG methods in respect of the computational cost. Since our method requires memory, the few amount of parameters increases. However, the inference time is competitive to other methods. Therefore, it can be said that our memory module is cost-effective with a high generalization score gain compared to the cost it occupies.

Visualization of memory activation. As illustrated in Fig. 6, we visualize the memory weight for the input image from the unseen domain. We can see that regions are activated by a memory slot corresponding to each class.

5. Conclusion and Future Work

We have presented the memory-guided meta-learning framework for robust semantic segmentation regardless of domain shift, with the novel memory divergence and feature cohesion losses. The ablation studies clearly demonstrate the effectiveness of each component and loss in our method. Finally, we have demonstrate that our method significantly outperforms other methods in domain generalization settings and also show competitive performance with domain adaptation methods. However, current DG methods (including this work) for semantic segmentation have limited to the assign the pixel corresponding to the underlying closed-set classes. To expand this work to a more practical scenario, we should consider the open-set segmentation which can be an appealing topic for DG in semantic segmentation. We remain a plethora of avenues for this as future work.

References

[1] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *ICCV*, 2021. 3

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2018. 12

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NIPS*, 2018. 3, 4

[4] Sergey Bartunov, Jack Rae, Simon Osindero, and Timothy Lillicrap. Meta-learning deep energy-based memory models. In *ICLR*, 2019. 3

[5] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. In *NIPS*, 2007. 1

[6] Jeffrey R Binder and Rutvik H Desai. The neurobiology of semantic memory. *TiCS*, 15(11):527–536, 2011. 2

[7] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NIPS*, 2011. 2

[8] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019. 1, 2

[9] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, 2019. 2

[10] 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. 4, 6, 12

[11] 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. 4, 6, 7, 12

[12] Wuyang Chen, Zhiding Yu, Shalini De Mello, Sifei Liu, Jose M Alvarez, Zhangyang Wang, and Anima Anandkumar. Contrastive syn-to-real generalization. In *ICLR*, 2020. 2

[13] Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Animashree Anandkumar. Automated synthetic-to-real generalization. In *ICML*, 2020. 2, 7

[14] 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. 2, 6, 7, 8, 12, 14, 15, 16, 17, 18, 19

[15] 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. 5, 14

[16] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NIPS*, 2019. 3

[17] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *ICCV*, 2019. 1, 2

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2, 8, 12

[19] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *ICLR*, 2018. 2

[20] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 2

[21] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016. 1

[22] Dayan Guan, Jiaxing Huang, Shijian Lu, and Aoran Xiao. Scale variance minimization for unsupervised domain adaptation in image segmentation. *PR*, 112:107764, 2021. 1

[23] Jianzhong He, Xu Jia, Shuaijun Chen, and Jianzhuang Liu. Multi-source domain adaptation with collaborative learning for semantic segmentation. In *CVPR*, 2021. 1, 2, 6, 7, 15

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7, 12

[25] 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. 1, 2, 7, 15

[26] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *ICCV*, 2021. 3

[27] Li Hu, Peng Zhang, Bang Zhang, Pan Pan, Yinghui Xu, and Rong Jin. Learning position and target consistency for memory-based video object segmentation. In *CVPR*, 2021. 3

[28] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *CVPR*, 2021. 2, 14

[29] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. Contextual-relation consistent domain adaptation for semantic segmentation. In *ECCV*, 2020. 1

[30] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. 1, 2

[31] Muireann Irish, Donna Rose Addis, John R Hodges, and Olivier Piguet. Considering the role of semantic memory in episodic future thinking: evidence from semantic dementia. *Brain*, 135(7):2178–2191, 2012. 2

[32] Muireann Irish and Olivier Piguet. The pivotal role of semantic memory in remembering the past and imagining the future. *Front. Behav. Neurosci.*, 7:27, 2013. 2

[33] 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, 14

[34] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Self-balanced learning for domain generalization. In *ICIP*, 2021. 3

[35] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, William T Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *NIPS*, 2018. 2

[36] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 1, 3, 4, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 19

[37] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Sequential learning for domain generalization. In *ECCV*, 2020. 3, 4

[38] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *ICCV*, 2019. 1, 3

[39] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018. 2

[40] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *ICML*, 2019. 3

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3, 4

[42] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2019. 1, 2

[43] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017. 2

[44] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 2

[45] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 6, 14

[46] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 3

[47] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 2, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 19

[48] Tiago Ramalho and Marta Garnelo. Adaptive posterior learning: few-shot learning with a surprise-based memory module. In *ICLR*, 2018. 3

[49] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2, 6, 14

[50] 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. 2, 6, 14

[51] 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. 3, 6, 12

[52] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 1

[53] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 3

[54] Daniel Saumier and Howard Chertkow. Semantic memory. *Curr. Neurol. Neurosci. Rep.*, 2(6):516–522, 2002. 2

[55] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Crossnorm and selfnorm for generalization under distribution shifts. In *ICCV*, 2021. 2, 7

[56] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019. 6

[57] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020. 1, 2

[58] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018. 5

[59] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *ECCV*, 2020. 1, 2

[60] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021. 3

[61] Zhonghua Wu, Xiangxi Shi, Guosheng Lin, and Jianfei Cai. Learning meta-class memory for few-shot semantic segmentation. In *ICCV*, 2021. 3

[62] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021. 3

[63] 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. 6, 14

[64] 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. 2, 7, 14

[65] Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Generalizable model-agnostic semantic segmentation via target-specific normalization. *PR*, 122:108292, 2022. 6, 7, 12, 15

[66] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021. 1, 2

[67] Qiming ZHANG, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *NIPS*, 2019. 1

[68] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019. 2

[69] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 12

[70] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *NIPS*, 2018. 2, 7, 15

[71] 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. 1, 2, 7, 15

[72] 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. 1, 2, 7, 15

[73] Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *CVPR*, 2021. 3, 5

[74] Xiantong Zhen, Yingjun Du, Huan Xiong, Qiang Qiu, Cees GM Snoek, and Ling Shao. Learning to learn variational semantic memory. In *NIPS*, 2020. 3

[75] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *IJCV*, 129(4):1106–1120, 2021. 1

[76] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 2

[77] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2020. 2

[78] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 1

[79] Andrea Zunino, Sarah Adel Bargal, Riccardo Volpi, Mehrnoosh Sameki, Jianming Zhang, Stan Sclaroff, Vittorio Murino, and Kate Saenko. Explainable deep classification models for domain generalization. In *CVPR*, 2021. 1, 2

Appendix

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. 7, 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 [14], 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+ [11] with ResNet-50 [24] backbone network. The output stride of DeepLabV3+ was set to 16 and adopted the auxiliary per-pixel cross-entropy loss proposed in PSPNet [69] with a coefficient of 0.4 to make a fair comparison with the normalization based DG method [14]. We performed memory operation using the feature map of 256 channel dimensions after the ASPP [11] module to leverage the multiple receptive fields and reduce GPU memory usage. We also adopted DeepLabV2 [10] with ResNet-101 for a fair comparison with multi-source unsupervised domain adaptation methods. For all the experiment, we initialized backbones with ImageNet [51] pre-trained model. The optimizer was SGD with momentum of 0.9. The learning rate of the meta-testing step

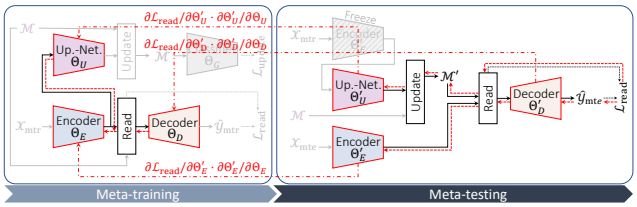


Figure 7. Illustration of the gradient flow (red dotted lines) in the optimization of meta-testing step.

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 [2, 18]. 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.

B.3. Re-implemented Methods

While IBN-Net [47] improved generalization ability by mixing instance normalization and batch normalization in the backbone, RobustNet [14] 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 [14][3]. To verify the effectiveness of our memory-guided meta-learning method, we re-implemented the MLDG [36] which is meta-learning based DG method. The augmentations and learning rates of MLDG were same with our method. Recently, TSMLDG [65] 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[4].

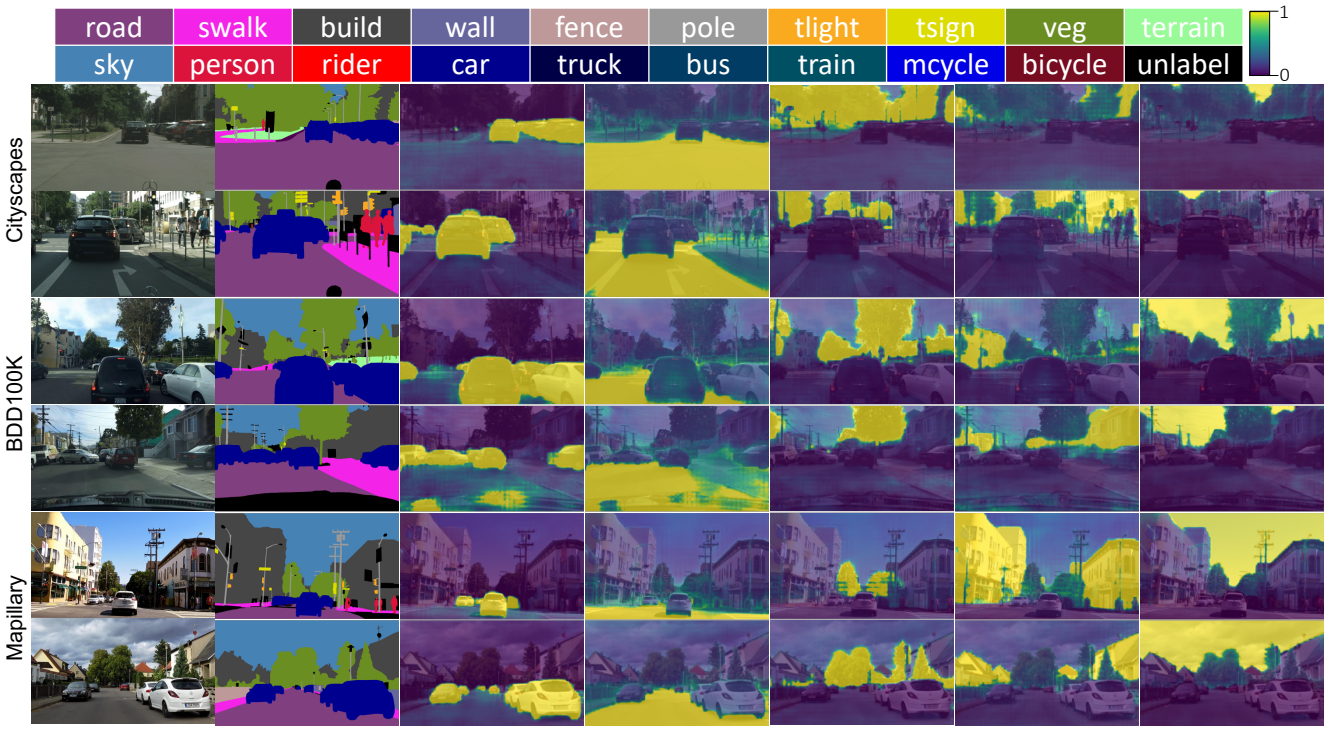
C. Additional Results

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 9, 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

[3]https://github.com/shachoi/RobustNet
[4]https://github.com/koncle/TSMLDG



(a) Images (b) Ground Truth (c) Car (d) Road (e) Vegetation (f) Building (g) Sky

Figure 8. **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.

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

Table 9. **Source (G+S)→Target (C, B, M):** Performance with or without memory updating network.

Methods	$\mathcal{L}_{\text{seg}}$	$\mathcal{L}_{\text{coh}}$	$\mathcal{L}_{\text{div}}$	Cityscapes	BDD100K	Mapillary	Avg.
IBN-Net [47]	✓	✗	✗	35.55	32.18	38.09	35.27
MLDG [36]	✓	✗	✗	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 10. **Source (G+S)→Target (C, B, M):** Mean IoU(%) comparison between the DG methods with only standard segmentation loss, $\mathcal{L}_{\text{seg}}$. All networks are DeepLabV3+ with ResNet50.

datasets in Fig. 8. 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 10. Without the proposed losses, our method still shows competitive performance against IBN-Net [47] and MLDG [36] 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.

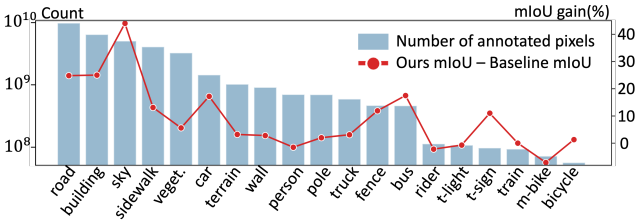


Figure 9. The correlation between the number of pixels per class in source datasets (G, S) and performance gain on BDD100K dataset.

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. 9, 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.

Methods	BDD100K	Mapillary	Avg.
MCIBI [33]	41.65	50.18	45.92
Ours	**46.78**	**55.10**	**50.94**

Table 11. **Source (C)→Target (B, M):** Mean IoU(%) comparison with MCIBI [33]. All networks are DeepLabV3 with ResNet50.

Backbone	Methods	Seg. model	Cityscapes	BDD100K	Mapillary
Resnet50	Baseline	FCN-8s	32.50	26.70	25.70
	DRPC [64]		37.40	32.10	34.10
	Baseline†	DeepLabV3+	29.00	25.10	28.20
	IBN-Net† [47]		33.90	32.30	37.80
	RobustNet† [14]		36.60	35.20	40.30
	Baseline		31.60	26.70	29.00
	MLDG‡ [36]		36.70	32.10	32.20
	Ours		41.00	34.60	37.40
Resnet101	FSDR [28]	DeepLabV2	44.75	39.66	40.87
	Ours		**44.90**	**39.71**	**41.31**

Table 12. **Source (G)→Target (C, B, M):** Mean IoU(%) comparison of other SOTA methods using various segmentation models and backbones. MLDG [36] is re-implemented. Results with † are from [14].

Comparison with MCIBI. We conduct comparison with MCIBI [33] 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 11, 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 12 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 [28] 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 13 and Table 14, respectively. Table 13 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 14, 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 [49] and Synthia [50]), and tested on the *unseen* real-world datasets [15, 45, 63].

In Fig. 10, we firstly conduct an additional comparison of the segmentation results on the Cityscapes [15] 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. 11 and Fig. 12 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 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. 11, 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. 12 shows the segmentation results under illumination and time changes. In addition, Fig. 12 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.

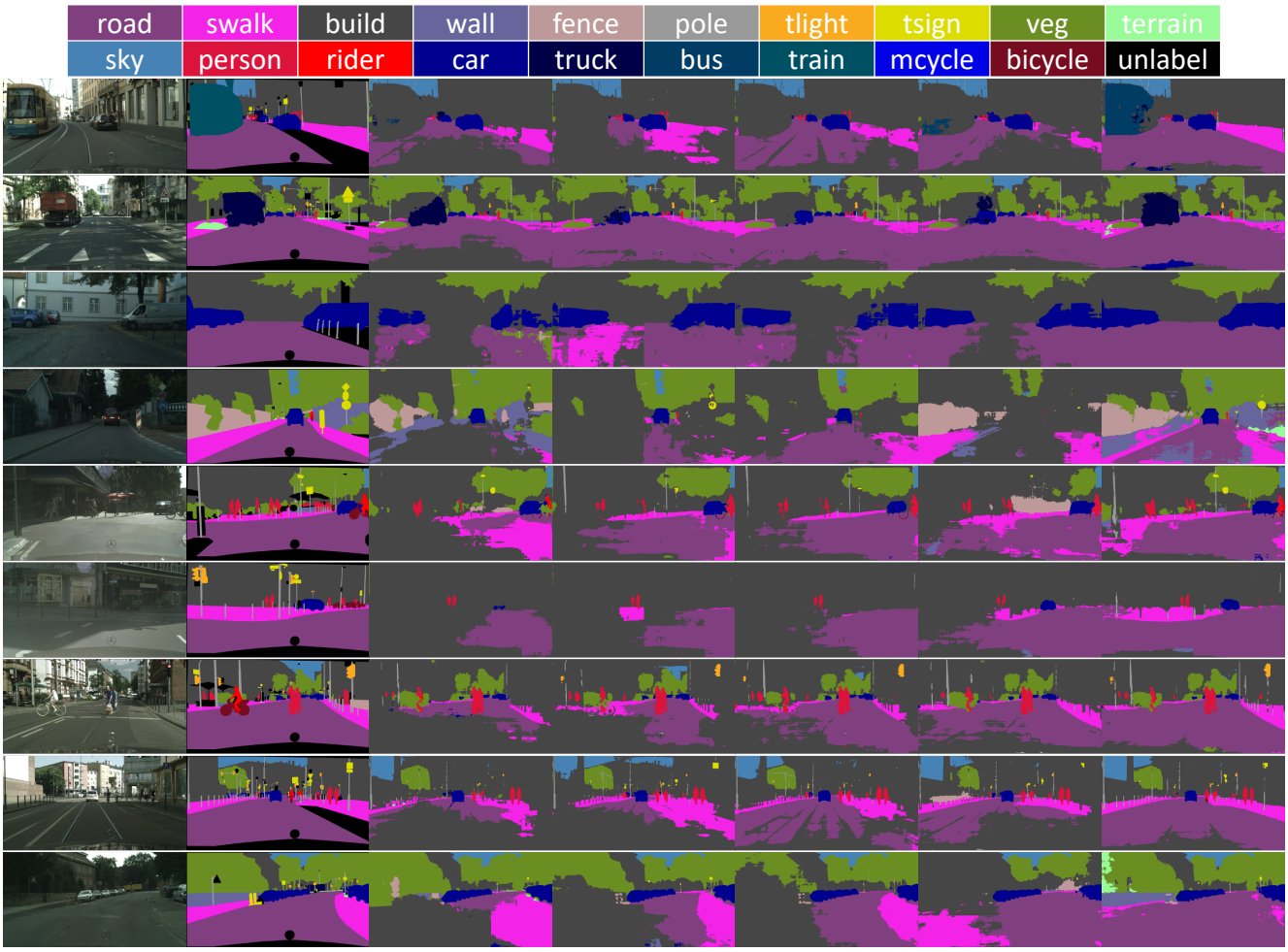
Finally, Fig. 13 and Fig. 14 show the segmentation results on the Mapillary dataset. The Mapillary dataset contains images acquired from various environments in Europe and Asia. Our method showed more reasonable results than other methods even when the viewpoint or scene structure changes in places such 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.

	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‡ [47]	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‡ [14]	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‡ [36]	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‡ [65]	**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‡ [47]	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‡ [14]	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‡ [36]	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‡ [65]	**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‡ [47]	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‡ [14]	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‡ [36]	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‡ [65]	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 13. **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 ‡.

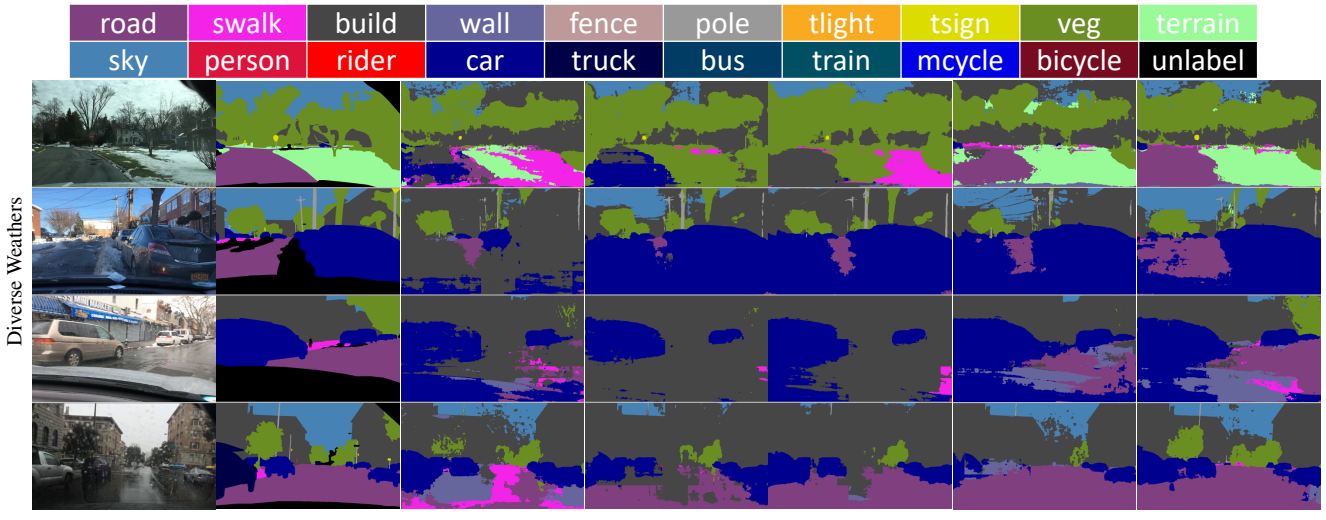
	Methods	w/Target	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegettion	sky	person	rider	car	bus	m-bike	bicycle	mIoU(%)
Cityscapes	Baseline	✗	77.1	32.4	75.3	13.8	11.5	29.0	13.7	10.3	81.5	79.1	53.1	10.2	80.2	39.0	21.9	11.5	40.0
	CyCADA† [25]	✓	86.8	41.4	74.7	15.5	3.4	27.3	3.8	0.2	73.2	72.4	51.9	12.7	82.7	41.8	18.5	23.3	39.3
	MDAN† [70]	✓	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† [72]	✓	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+† [71]	✓	**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 [23]	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**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† [25]	✓	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† [70]	✓	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† [72]	✓	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+† [71]	✓	**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 14. **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 [71].



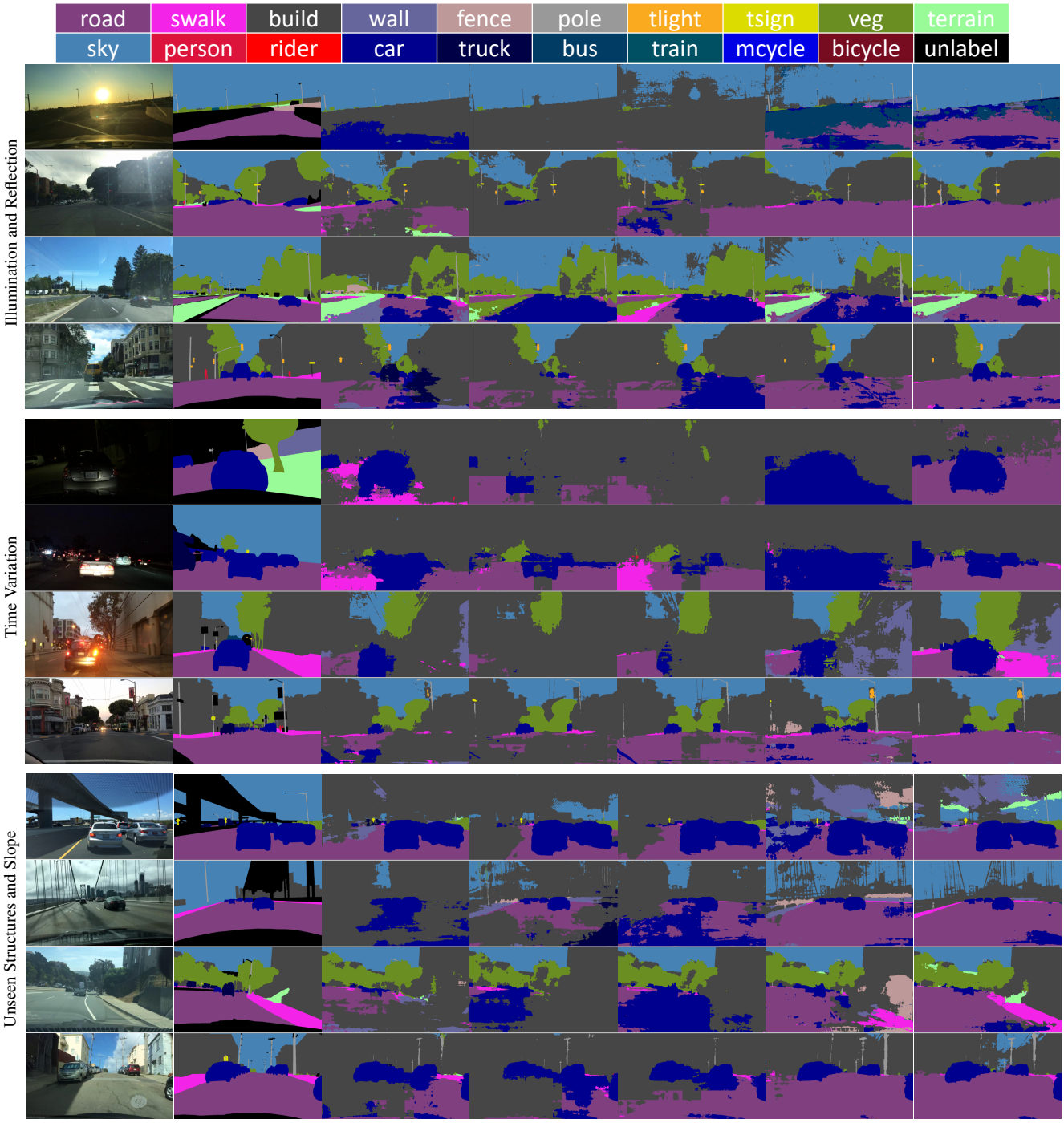
(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [47] (e) RobustNet [14] (f) MLDG [36] (g) Ours

Figure 10. **Source (G+S)→Target (C):** Qualitative comparison on the Cityscapes dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)



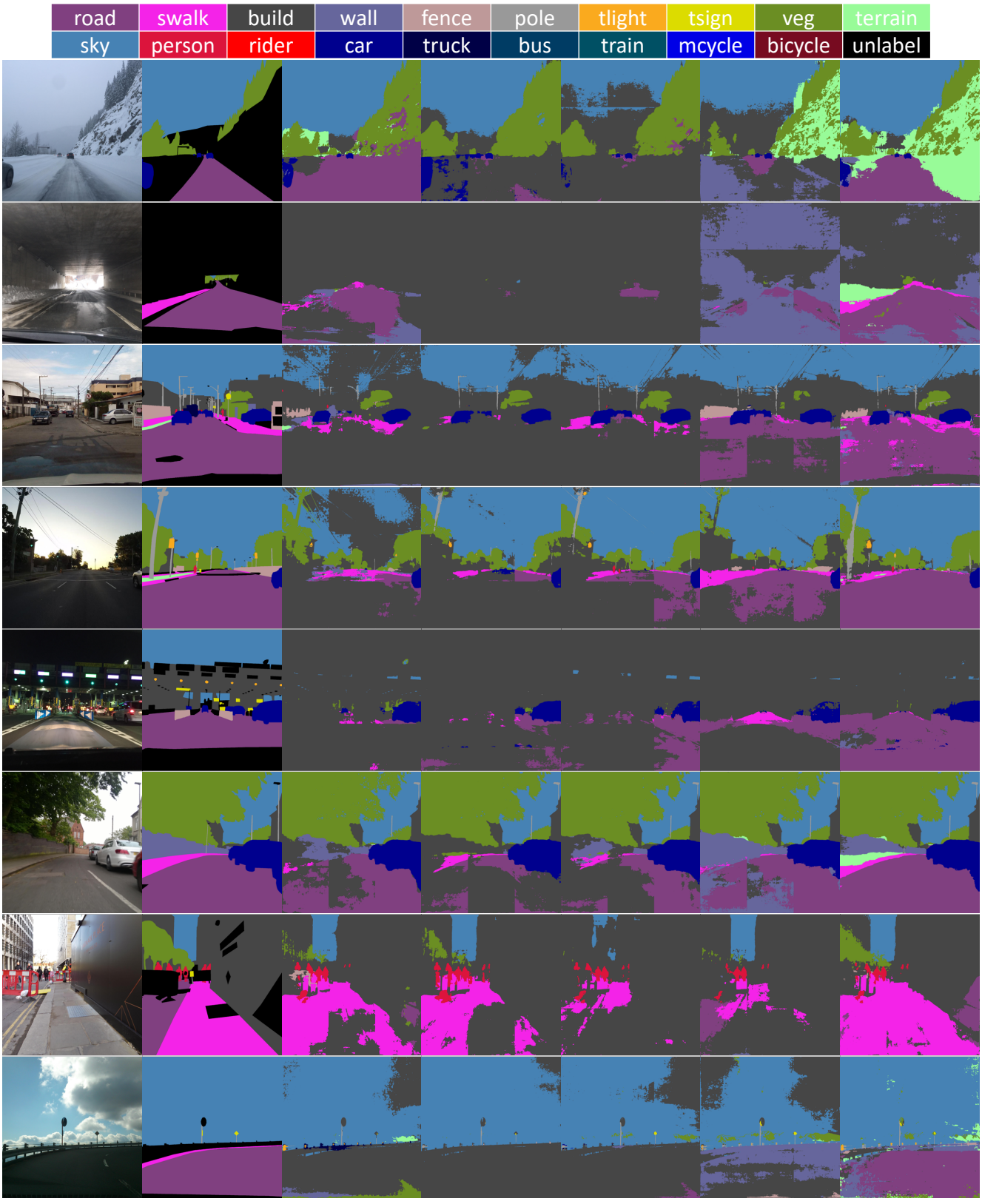
(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [47] (e) RobustNet [14] (f) MLDG [36] (g) Ours

Figure 11. **Source (G+S)→Target (B):** [1/2] Qualitative comparison on the BDD100K dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)



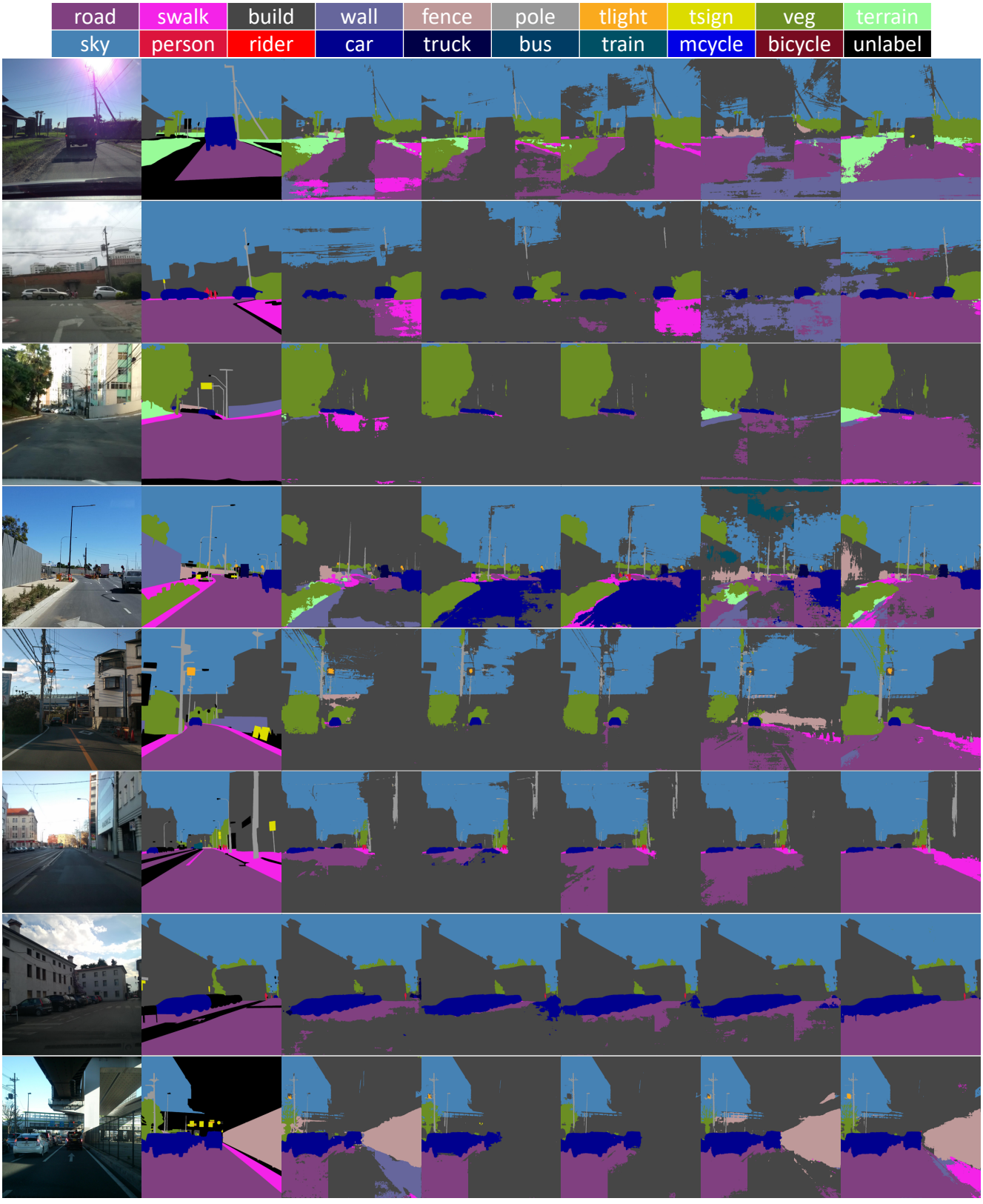
(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [47] (e) RobustNet [14] (f) MLDG [36] (g) Ours

Figure 12. **Source (G+S)→Target (B):** [2/2] Qualitative comparison on the BDD100K dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)



(a) Images (b) Ground Truth (c) Baseline (d) IBN-Net [47] (e) RobustNet [14] (f) MLDG [36] (g) Ours

Figure 13. **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 [47] (e) RobustNet [14] (f) MLDG [36] (g) Ours

Figure 14. **Source (G+S)→Target (M):** [2/2] Qualitative comparison on the Mapillary dataset. All methods adopt DeepLabV3+ with ResNet50. (Best viewed in color.)