Open Compound Domain Adaptation with Object Style Compensation for Semantic Segmentation

Tingliang Feng[1†] **Hao Shi**[1,2†] **Xueyang Liu**[1] **Wei Feng**[1]
Liang Wan[1] **Yanlin Zhou**[3] **Di Lin**[1*]
[1]College of Intelligence and Computing, Tianjin University
[2]Department of Automation, Tsinghua University [3]Dunhuang Academy
{fengtl, xyliu850569498, lwan}@tju.edu.cn shi-h23@mails.tsinghua.edu.cn
wfeng@ieee.org zhouyanlin@dha.ac.cn Ande.lin1988@gmain.com

Abstract

Many methods of semantic image segmentation have borrowed the success of open compound domain adaptation. They minimize the style gap between the images of source and target domains, more easily predicting the accurate pseudo annotations for target domain's images that train segmentation network. The existing methods globally adapt the scene style of the images, whereas the object styles of different categories or instances are adapted improperly. This paper proposes the *Object Style Compensation*, where we construct the *Object-Level Discrepancy Memory* with multiple sets of discrepancy features. The discrepancy features in a set capture the style changes of the same category's object instances adapted from target to source domains. We learn the discrepancy features from the images of source and target domains, storing the discrepancy features in memory. With this memory, we select appropriate discrepancy features for compensating the style information of the object instances of various categories, adapting the object styles to a unified style of source domain. Our method enables a more accurate computation of the pseudo annotations for target domain's images, thus yielding state-of-the-art results on different datasets.

1 Introduction

The recent methods [1, 2, 3, 4, 5] of semantic segmentation utilize the open compound domain adaptation (OCDA), which harnesses the annotated images and the annotation-free images captured in the open environments to train the segmentation network. In this manner, the segmentation network learns from richer data with diverse object appearances while requiring reasonable effort for image labeling. Here, the annotated images belong to the source domain, while the annotation-free images are in the compound target domain for capturing the complexity of open environments.

There is a gap between the image styles of the source and target domains, where the image styles usually are regarded as an array of scene-level properties (e.g., weather and lighting conditions). The scene styles are adapted[2] to narrow the gap between image styles, allowing the segmentation network to focus on the intrinsic object appearances of every domain. It helps the network to predict accurate segmentation masks for many images in the compound target domain. The predicted masks play as the pseudo-pixel-wise annotations for tuning the segmentation network.

Typically, the adaptation of scene styles follows the homologous patterns to change the object styles in the same image. It lets various object categories undergo the style change along the same direction (see Figure 1(a), the clear pedestrians and overexposed cars adapted from the night to day scenes). In the common context, the objects in various categories, even different object instances in the same category, may have specific styles within a scene (see Figure 1(b), the usual cars and pedestrians in the day time). Adapting scene styles may yield unreasonable object styles inconsistent with the usual context, leading to problematic annotations for the network training.

*Di Lin is the corresponding author of this paper.
[2]Someone performs the adaptation in the image or feature space. The latter one is considered in this paper.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

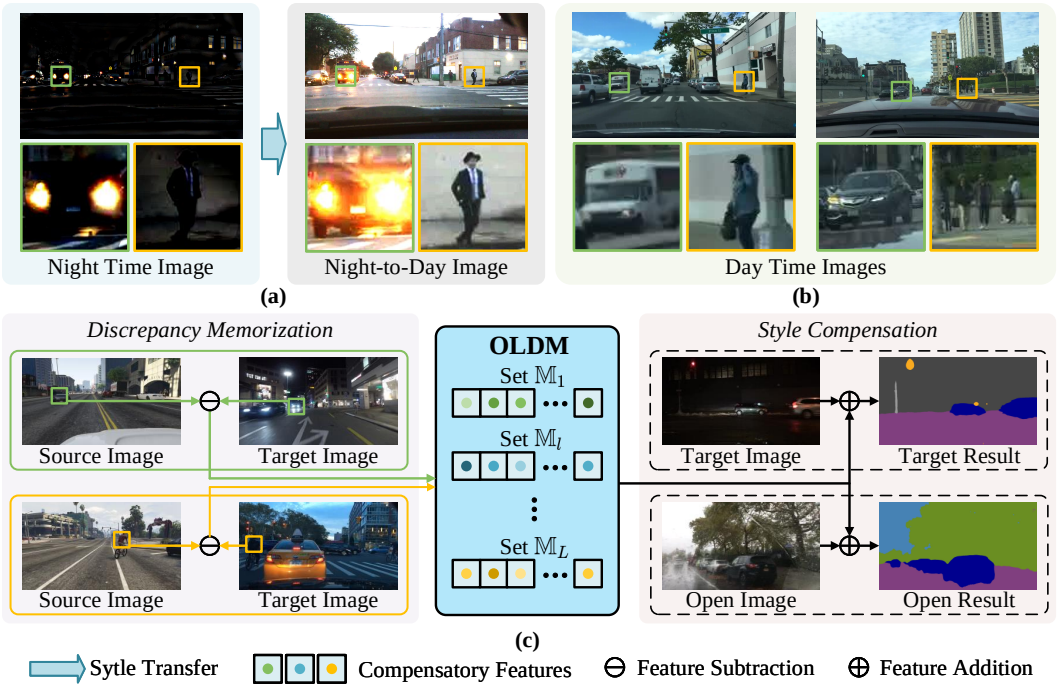

Figure 1: (a) Cars and pedestrians adapted from the night to day scenes. (b) The usual cars and pedestrians in the real day scene. (c) OLDM contains the discrepancy features for representing the difference of object styles in the source and target domains, which are used for adapting the object-level styles from target to source domains.

This paper advocates equipping OCDA with object style compensation for semantic segmentation. In contrast to the adaptation of scene styles, the compensation respects the object style discrepancies between the source and target domains. These discrepancies are captured for the independent object categories and instances. Intuitively, the style discrepancies can be memorized during network learning. It enables the appropriate style discrepancies, which can be regarded as the prior information, to be selected and added to the object features. Consequently, we compensate the object features to adapt the object styles to the similar style of source domain. The compensation yields consistent object context within the scene, helping the segmentation network to compute reliable pseudo annotations.

Specifically, we conduct the object style compensation for adapting the images from target to source domains. The pipeline comprises *Discrepancy Memorization* and *Style Compensation*, as illustrated in Figure 1(c). During *Discrepancy Memorization*, we construct a feature base, *Object-Level Discrepancy Memory* (OLDM), which consists of multiple feature sets. Each set contains the discrepancy features for the identical object category. Here, we compute the discrepancy feature by subtracting the target domain's object features from the same category's object features of the source domain, letting the discrepancy features represent the difference of the object styles across two domains. We compute the discrepancy features for the object instances in different images of target domain. During *Style Compensation*, given a query object in the target domain's image, we select the discrepancy feature from the feature set of the requested category, where the discrepancy feature represents the instance much relevant to the query object. In this way, the discrepancy feature customizes the information, which is category- and instance-orientated, for compensating the query object's feature. The compensated object features are used for regressing the pseudo annotations.

We conduct an intensive evaluation of the object style compensation. On the public datasets (e.g., C-Driving [1], ACDC [6], Cityscapes [7], KITTI [8], and WildDash [9]) that allows OCDA to assist semantic segmentation, the object style compensation surpasses state-of-the-art methods, demonstrating its effectiveness.

2 Related Work

2.1 Domain Adaptation for Semantic Segmentation

There have been many works on the unsupervised domain adaptation (UDA) [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21], the multi-target domain adaptation (MTDA) [22, 23, 23, 23, 24, 25] and the

domain generalization (DG) [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37] for semantic segmentation. These methods either consider the setting of single or multi-target domain adaptation or domain generalization. In contrast, OCDA achieves both domain adaptation and domain generalization and better domain adaptation performance. Liu et al. [1] propose the setting of OCDA for handling unlabeled compound target domain and open domain. Park et al. [2] reformulate the complex OCDA problem as multiple UDA problems. Gong et al. [3] propose a principled meta-learning-based OCDA for semantic segmentation. Pan et al. [5] propose a multi-teacher framework with bidirectional photometric mixing to adapt to every target domain. Kundu et al. [4] propose the amplitude spectrum transformation in the feature space for OCDA.

The existing methods adapt scene styles to fill in the gap of image appearances between different domains. They are insensitive to the object styles, which are critical to capture the style change of the object appearances of various categories/instances. In contrast, we propose object-style compensation by respecting the individual categories/instances. Our approach appropriately adapts the object styles to the styles similar to the source domain and yield consistent context in the same scene.

2.2 Deep Network with Memorization for Visual Understanding

Recent studies demonstrate the importance of the deep network with memorization for visual understanding [38, 39, 40, 41, 42, 43, 44, 44, 45, 46, 47, 48, 49]. Wang et al. [50] explore an ample data space for memorizing the category-level information. Liang et al. [51] present a meta-learning framework that leverages memory-based guidance to capture and retain the co-occurring categorical knowledge shared among objects of the same category across different domains. VS et al. [52] propose memory-guided attention to incorporate category information into the domain adaptation process. Yang et al. [53] propose a hybrid quality-aware triplet memory to improve the quality and stability of generated pseudo labels. Kim et al. [54] present a memory-guided semantic segmentation method that abstracts the conceptual knowledge of semantic classes into the memory.

The previous methods utilize memory that stores scene-level or category-level features extracted from the images. Because these features mainly capture a single domain's image style, they are less powerful for adapting a broad range of image styles of the compounded target domain to the source domain. In contrast, we propose the external memory to store both category- and instance-level discrepancy features learned across different domains. These discrepancy features can directly compensate the images of the target domains, whose category- and instance-level styles are adapted to the source domain. The alignment of the object styles in different domains' images helps the segmentation network to focus on the intrinsic object appearances for producing the pseudo labels.

3 Method Overview

We illustrate the object style compensation in Figure 2. $\mathbf{I}_s, \mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$ are the images in the source and target domains. $\mathbf{I}_s$/$\mathbf{I}_t$ is associated with/without the pixel-wise annotation. We denote $\mathbf{Y}_s \in \mathbb{R}^{H\times W\times L}$ (L is the category number) as the ground-truth annotation for $\mathbf{I}_s$. The encoder extracts the object feature maps $\mathbf{F}_s, \mathbf{F}_t \in \mathbb{R}^{H\times W\times C}$ from $\mathbf{I}_s, \mathbf{I}_t$. C indicates the feature channels. $\mathbf{F}_s(x,y), \mathbf{F}_t(x,y) \in \mathbb{R}^C$ represent the object located at (x,y) in the source and target images.

The object style compensation includes *Discrepancy Memorization* and *Style Compensation*. The discrepancy memorization uses $\mathbf{F}_s, \mathbf{F}_t$ to construct the *Object-Level Discrepancy Memory* (OLDM). OLDM stores the discrepancy features, which capture the change of object styles from target to source domain. For the target image $\mathbf{I}_t$, the style compensation uses discrepancy features to adapt the object style information of $\mathbf{F}_t$, yielding the compensated feature map $\widetilde{\mathbf{F}}_t \in \mathbb{R}^{H\times W\times C}$ used by decoder to compute the pseudo annotation $\mathbf{Y}_t \in \mathbb{R}^{H\times W\times L}$.

Discrepancy Memorization The discrepancy memorization only takes place during network training. Here, we construct category-key and OLDM as $\{(\mathbf{A}_l, \mathbb{M}_l) \mid \mathbf{A}_l \in \mathbb{R}^C, l = 1, ..., L\}$, which contains pairs of category-key feature (e.g., $\mathbf{A}_l$) and feature set (e.g., $\mathbb{M}_l$). $\mathbf{A}_l$ is the category-key feature that captures the source domain's representative style of the l^{th} category. The set $\mathbb{M}_l = \{(\mathbf{N}_{l,m}, \mathbf{D}_{l,m}) \mid \mathbf{N}_{l,m}, \mathbf{D}_{l,m} \in \mathbb{R}^C, m = 1, ..., M\}$ has pairs of instance-key feature (e.g., $\mathbf{N}_{l,m}$) and discrepancy feature (e.g., $\mathbf{D}_{l,m}$). $\mathbf{N}_{l,m}$ captures a representative instance-level style of the l^{th} category in target domain. We associate $\mathbf{N}_{l,m}$ with the discrepancy feature $\mathbf{D}_{l,m}$, which captures the change of the representative instance-level style from target to source domain.

We illustrate the discrepancy memorization in Figure 3(a), where we update the category-key, instance-key, and discrepancy features. We use the l^{th} category's object features in $\mathbf{F}_s$ to update the category-key feature $\mathbf{A}_l$. In the set $\mathbb{M}_l$, we measure the similarities between the l^{th} category's object

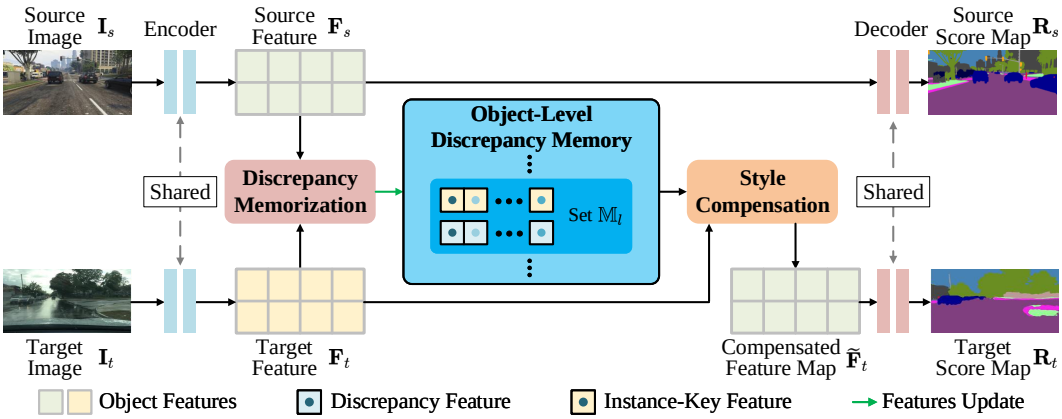

Figure 2: Overview of the object style compensation. In discrepancy memorization, we learn the discrepancy features from different categories and instances of object features of the source and target images. The discrepancy features are stored in the object-level discrepancy memory. In style compensation, we select the discrepancy features from memory for compensating the object features of the target domain. Based on the compensated features, we regress the segmentation score map of the target image, which plays as the pseudo annotation for updating the segmentation network.

features in $\mathbf{F}_t$ and the instance-key features $\{\mathbf{N}_{l,m} \mid m = 1, ..., M\}$. According to these similarities, we use the l^{th} category's object feature $\mathbf{F}_t(x, y)$ updates instance-key features, while calculating the difference between $\mathbf{A}_l$ and $\mathbf{F}_t(x, y)$ to update the discrepancy features $\{\mathbf{D}_{l,m} \mid m = 1, ..., M\}$.

Style Compensation We conduct the style compensation during training and testing. We illustrate the style compensation in Figure 3(b), where we use the discrepancy features in OLDM to compensate the target image's object feature map $\mathbf{F}_t$. Given the object feature $\mathbf{F}_t(x, y)$ as a query, we use a segmentation head to regress its category scores $\mathbf{R}_t(x, y) \in \mathbb{R}^{H\times W\times L}$, selecting the feature sets $\{\mathbb{M}_1, ..., \mathbb{M}_K\}$ of OLDM for the style compensation. In $\mathbb{M}_k$, we calculate the similarities between $\mathbf{F}_t(x, y)$ and the instance-keys $\{\mathbf{N}_{k,m} \mid m = 1, ..., M\}$. These similarities weight the discrepancy features $\{\mathbf{D}_{k,m} \mid m = 1, ..., M\}$. After averaging the weighted discrepancy features of each set, the results are added to $\mathbf{F}_t(x, y)$, yielding the compensated feature $\widetilde{\mathbf{F}}_t(x, y)$. $\widetilde{\mathbf{F}}_t$ is fed into the decoder for regressing the pseudo annotation $\mathbf{Y}_t$. We borrow the pseudo annotation $\mathbf{Y}_t$ paired with the target image $\mathbf{I}_t$ to fine-tune the segmentation network.

4 Object Style Compensation

4.1 Discrepancy Memorization

During network training, we employ discrepancy memorization to construct the *Object-Level Discrepancy Memory* (OLDM). We denote category-key and OLDM as $\{(\mathbf{A}_l, \mathbb{M}_l) \mid \mathbf{A}_l \in \mathbb{R}^C, l = 1, ..., L\}$, where $\mathbb{M}_l = \{(\mathbf{N}_{l,m}, \mathbf{D}_{l,m}) \mid \mathbf{N}_{l,m}, \mathbf{D}_{l,m} \in \mathbb{R}^C, m = 1, ..., M\}$. We use the feature maps $\mathbf{F}_s, \mathbf{F}_t \in \mathbb{R}^{H\times W\times C}$, which are extracted from the source and target images $\mathbf{I}_s, \mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$ by encoder, to update the category-key, instance-key, and discrepancy features in OLDM (see Figure 3(a)).

Update of Category-Key Features Given the feature map $\mathbf{F}_s$ of the source image $\mathbf{I}_s$, we update the category-key features $\{\mathbf{A}_l \in \mathbb{R}^C \mid l = 1, ..., L\}$. We employ decoder to predict a category for every object feature in the map $\mathbf{F}_s$. The source image $\mathbf{I}_s$ has the ground-truth annotation $\mathbf{Y}_s \in \mathbb{R}^{H\times W\times L}$. The decoder predicts the category scores $\mathbf{R}_s(x, y) \in \mathbb{R}^L$, where $\mathbf{R}_s(x, y, l) \in \mathbb{R}$ is the score for predicting the object located at (x, y) of the source image to be the l^{th} category. We assume the category l leads to the highest score $\mathbf{R}_s(x, y, l)$. The predicted label is compared to the ground-truth label $\mathbf{Y}_s(x, y)$. The correct prediction means the object feature $\mathbf{F}_s(x, y)$ is reliable for updating the category-key feature $\mathbf{A}_l$ as:

$$\begin{aligned} \mathbf{A}_l &\leftarrow \mathbf{A}_l + \lambda \mathbf{F}_s(x, y), \\ s.t.\ \ \mathbf{R}_s(x, y, l) &= \max\{\mathbf{R}_s(x, y, 1), ..., \mathbf{R}_s(x, y, L)\}, \quad l = \mathbf{Y}_s(x, y), \end{aligned} \tag{1}$$

where $\leftarrow$ means the update by overwriting. λ is a ratio for controlling the information of $\mathbf{F}_s(x, y)$ injected into $\mathbf{A}_l$. In Eq.(1), $\mathbf{A}_l$ aggregates the information of an array of object features, which

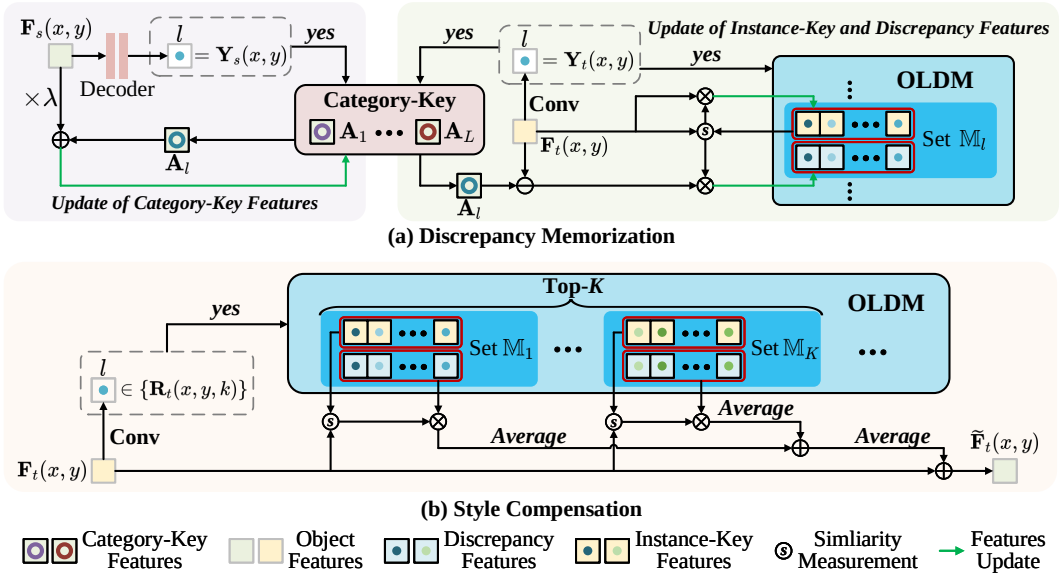

Figure 3: (a) In the discrepancy memorization, we use the object features of the source and target domains, along with their differences, to update the category-key, instance-key, and discrepancy features in OLDM. (b) Given the object features of the target domain, we compute the similarities between the instance-key features and each object feature of the target domain, which is compensated by the discrepancy features weighted by the similarities.

consistently point to the l^{th} category. It helps $\mathbf{A}_l$ to capture the representative object style of the l^{th} category in source domain, which is used to compute discrepancy features.

Update of Instance-Key and Discrepancy Features We use the feature map $\mathbf{F}_t$ of the target image $\mathbf{I}_t$ to update the instance-key features in OLDM. For the object feature $\mathbf{F}_t(x,y) \in \mathbb{R}^C$, we use the intermediate segmentation head to regress the category scores in $\mathbf{R}_t(x,y) \in \mathbb{R}^L$, where $\mathbf{R}_t(x,y,l) \in \mathbb{R}$ is the score for predicting the object located at (x,y) of the target image to be the l^{th} category. We use $\mathbf{F}_t(x,y)$ to update the instance-key feature $\mathbf{N}_{l,m}$ as:

$$\mathbf{N}_{l,m} \leftarrow \mathbf{N}_{l,m} + \mathbf{w}_{l,m}\mathbf{F}_t(x,y), \quad \mathbf{D}_{l,m} \leftarrow \mathbf{D}_{l,m} + \mathbf{w}_{l,m}(\mathbf{A}_l - \mathbf{F}_t(x,y)),$$
$$s.t. \quad \mathbf{R}_t(x,y,l) = \max\{\mathbf{R}_t(x,y,1), ..., \mathbf{R}_t(x,y,L)\} > \gamma; \quad \mathbf{w}_{l,m} = \frac{\mathbf{N}_{l,m} \cdot \mathbf{F}_t(x,y)}{\sqrt{C}}. \tag{2}$$

The category l has the highest score in $\{\mathbf{R}_t(x,y,1), ..., \mathbf{R}_t(x,y,L)\}$. We classify the object feature $\mathbf{F}_t(x,y)$ to the l^{th} category, thus updating the instance-key features $\{\mathbf{N}_{l,m} \mid m = 1, ..., M\}$ in the set $\mathbb{M}_l$ for the same category. $\mathbf{w}_{l,m} \in \mathbb{R}$ is a weight, which is calculated as the dot-product between the instance-key feature $\mathbf{N}_{l,m}$ and the object feature $\mathbf{F}_t(x,y)$, whose similarity is measured. A high weight lets object feature contribute more information to instance-key feature.

Unlike decoder supervised by the ground-truth annotations of source images, the intermediate segmentation head is supervised by the pseudo annotations of target images. It may misclassify object feature, thus misleading the update of instance-key features. To remedy this issue, we set a score threshold γ in Eq. (2), to select the reliable object feature $\mathbf{F}_t(x,y)$, whose highest score $\mathbf{R}_t(x,y,l)$ is greater than γ. This threshold only allows the object feature, whose category scores are confident, to update instance-key features. Each instance-key feature aggregates the features of various object instances with similar styles.

In Eq (2), we also update the discrepancy features $\{\mathbf{D}_{l,m} \in \mathbb{R}^C \mid m = 1, ..., M\}$. Here, we subtract the object feature $\mathbf{F}_t(x,y)$ from the category-key feature $\mathbf{A}_l$. Note that $\mathbf{F}_t(x,y)$ and $\mathbf{A}_l$ are extracted from the target and source images, respectively. Thus, the subtraction of these features captures the difference between the object styles of in target and source images, which is weighted by $\mathbf{w}_{l,m}$ to add to the discrepancy feature $\mathbf{D}_{l,m} \in \mathbb{R}^C$.

We remark that the update of OLDM is disabled during network testing. The style compensation uses the most updated category-key, instance-key, and discrepancy features in OLDM.

4.2 Style Compensation

The style compensation works during network training and testing. It harnesses the category- and instance-key features to find the appropriate discrepancy features in OLDM, which compensate for the object features of target image (see Figure 3(b)).

Compensation of Object Features Given the object feature $\mathbf{F}_t(x, y)$ as a query, which is extracted from the target image $\mathbf{I}_t$, we use the intermediate segmentation head to predict the category scores $\mathbf{R}_t(x, y)$. Next, we select the instance-key and discrepancy features from $\{\mathbb{M}_1, ..., \mathbb{M}_K\}$, where $\mathbb{M}_k = \{(\mathbf{N}_{k,m}, \mathbf{D}_{k,m}) \mid \mathbf{N}_{k,m}, \mathbf{D}_{k,m} \in \mathbb{R}^C, m = 1, ..., M\}$ to compensate the object feature $\mathbf{F}_t(x, y)$, yielding the compensated feature $\widetilde{\mathbf{F}}_t(x, y) \in \mathbb{R}^C$ as:

$$\widetilde{\mathbf{F}}_t(x, y) = \mathbf{F}_t(x, y) + \sum_k \sum_m \mathbf{w}_{k,m} \mathbf{D}_{k,m},$$

$$s.t.\ \mathbf{R}_t(x, y, k) \in \max{}_K\{\mathbf{R}_t(x, y, 1), ..., \mathbf{R}_t(x, y, L)\};\ \mathbf{w}_{k,m} = \frac{\mathbf{N}_{k,m} \cdot \mathbf{F}_t(x, y)}{\sqrt{C}}, \tag{3}$$

where $\max_K$ means to select the top-K relevant categories. We use Eq. (3) to achieve the compensated feature $\widetilde{\mathbf{F}}_t \in \mathbb{R}^{H \times W \times C}$, which is used for computing the pseudo annotation for the target image $\mathbf{I}_t$.

Computation of Pseudo Annotations Based on the compensated feature $\widetilde{\mathbf{F}}_t$, the decoder of the segmentation network predicts the category score map $\widetilde{\mathbf{R}}_t \in \mathbb{R}^{H \times W \times L}$. We use the category score map $\widetilde{\mathbf{R}}_t$ to produce the pseudo annotation $\mathbf{Y}_t \in \mathbb{R}^{H \times W \times L}$, where the pixel-wise annotation $\mathbf{Y}_t(x, y) \in \mathbb{R}^L$ for the pixel located at (x, y) is determined as:

$$\mathbf{Y}_t(x, y, l) = \begin{cases} 1 & \max\{\widetilde{\mathbf{R}}_t(x, y, 1), ..., \widetilde{\mathbf{R}}_t(x, y, L)\} = \widetilde{\mathbf{R}}_t(x, y, l) > \gamma, \\ 0 & \max\{\widetilde{\mathbf{R}}_t(x, y, 1), ..., \widetilde{\mathbf{R}}_t(x, y, L)\} > \max\{\widetilde{\mathbf{R}}_t(x, y, l), \gamma\}, \\ \text{ignored} & \gamma \geq \max\{\widetilde{\mathbf{R}}_t(x, y, 1), ..., \widetilde{\mathbf{R}}_t(x, y, L)\}. \end{cases} \tag{4}$$

The pixel-wise annotation $\mathbf{Y}_t(x, y)$ is a one-hot vector. We set the l^{th} channel $\mathbf{Y}_t(x, y, l)$ to 1 (see the first case), when the category score $\widetilde{\mathbf{R}}_t(x, y, l)$ is higher than other scores in the set $\{\widetilde{\mathbf{R}}_t(x, y, 1), ..., \widetilde{\mathbf{R}}_t(x, y, L)\}$; otherwise, we set $\mathbf{Y}_t(x, y, l)$ to 0 (see the second case). It should be noted that the threshold γ is used in the third case, where too low scores let the pseudo annotations be ignored during network training. The computation of pseudo annotations only takes place during network training. For network testing, we resort to the category score map $\widetilde{\mathbf{R}}_t$ to predict the labels for all pixels, following the convention of semantic segmentation.

4.3 Supervision for Network Training

We use the ground-truth annotations of source images and the pseudo annotations of target images to train segmentation network. We formulate the overall training objective $\mathcal{L}$ as:

$$\mathcal{L} = \mathcal{L}_{gt} + \mathcal{L}_{pse}. \tag{5}$$

$\mathcal{L}_{gt}$ and $\mathcal{L}_{pse}$ represent the objectives with the supervision of ground-truth and pseudo annotations.

Supervision of Ground-truth Annotations Based on the object feature map $\mathbf{F}_s$ of the source image $\mathbf{I}_s$, decoder predicts the category score map $\mathbf{R}_s$. We use the cross-entropy loss $\mathcal{CE}$ to penalize the difference between the predicted score map $\mathbf{R}_s$ and the ground-truth annotation $\mathbf{Y}_s$ as:

$$\mathcal{L}_{gt} = \mathcal{CE}(\mathbf{R}_s, \mathbf{Y}_s). \tag{6}$$

Supervision of Pseudo annotations We use intermediate segmentation head to regress the category score map $\mathbf{R}_t$ for the target image $\mathbf{I}_t$, based on the object feature map $\mathbf{F}_t$. We compute the cross-entropy loss, which penalizes the difference between the score map $\mathbf{R}_t$ and the pseudo annotation $\mathbf{Y}_t$ as the first term of the below Eq. (7).

$$\mathcal{L}_{pse} = \mathcal{CE}(\mathbf{R}_t, \mathbf{Y}_t) + \mathcal{CE}(\widetilde{\mathbf{R}}_t, \mathbf{Y}_t). \tag{7}$$

Moreover, we leverage the decoder to predict the category score map $\widetilde{\mathbf{R}}_t$, based on the compensated feature map $\widetilde{\mathbf{F}}_t$. We again use cross-entropy loss (see the second term of Eq. (7)) to measure the segmentation errors in the score map $\widetilde{\mathbf{R}}_t$, by comparing to the pseudo annotation $\mathbf{Y}_t$.

5 Experiments

5.1 Experimental Datasets

We use GTA5 [55], SYNTHIA [56], C-Driving [1], ACDC [6], Cityscapes [7], KITTI [8], and WildDash [9] datasets to evaluate our method. The images in a dataset may be subdivided into source, target, and open domains. All images with annotations in source domain and a portion of images without annotations in target domain are used for network training. We evaluate the segmentation performances on the rest images in target domain and all images in open domain. We list the division of these datasets in Table 1.

Table 1: Divisions of the experimental datasets.

Dataset	Total	Train		Test
		Source	Target	Open
GTA5	24,966	24,966	-	-
SYNTHIA	9,400	9,400	-	-
C-Driving	16,127	-	14,697	803	627
ACDC	1,906	-	1,200	306	400
Cityscapes	500	-	-	-	500
KITTI	200	-	-	-	200
WashDash	638	-	-	-	638

In the alation study of our method, we use 24,966 source images of GTA5 dataset and 14,697 target images of C-Driving dataset for network training. 803 and 627 images of target and open domains in C-Driving dataset are used for testing. We report the segmentation performance regarding mean intersection-over-unions (mIoUs) on target and open domains.

5.2 Ablation Study

Analysis of Discrepancy Memorization In Tables 2, 3, and 4, we study various strategies for updating category-key, instance-key, and discrepancy features during the discrepancy memorization.

We report different strategies for using category-key features in Table 2. First, we disable the update of category-key features. This is done by removing OLDM (see "w/o OLDM") during training and testing, producing the performance of the stand-alone segmentation network. Another alternative uses the mean of the instance-key features in the same set to replace category-key feature but yields lower performances than our method (see "mean instances"). This is because category-key and instance features are computed based on the image features of discrepant domains (i.e., source and target domains), making none of them replaceable. Second, we experiment with using the mean of the image features of source domain in a local mini-batch to compute the category-key features, which are overridden by new mini-batch. This strategy achieves worse results than our method (see "local"), because we use different mini-batches to compute category-key features globally (see "global"), which more comprehensively capture the object features of source domain.

Table 2: Results of various ways of using category-key features. mIoU(T) and mIoU(O) mean the mIoUs on target and open domains.

Category-Key	Method	mIoU(T)	mIoU(O)
✘	w/o OLDM	36.6	39.7
	mean instances	39.2	41.5
✔	local	41.7	43.2
	global	**44.1**	**46.9**

In Table 3, we compare the performances of various strategies for using instance-key features. By eliminating the instance-key features in OLDM, we lack the similarities between the object features of target domain and the instance-key features for weighting the discrepancy features to compensate for object features. In this case, we experiment with simply averaging discrepancy features (see "mean discrepancy"), or directly computing the similarities between object features and discrepancy features (see "discrepancy similarity") for compensation. Without instance-key features, these naive strategies lead to worse results than our method. This is because the average discrepancy features and the alternative similarities inappropriately match object instances to the discrepancy features desired for compensation. We also experiment with enabling and updating instance-key features in various ways. For each set of instance-key and discrepancy features of the same category, we select the instance-key feature, which is updated along with associated discrepancy feature by the object feature of target domain. It degrades the performances (see "top-1 update"), compared to adequately updating 50% or 100% of instance-key and discrepancy features (see "top-50% update" and "100% update").

Table 3: Results of various ways of using instance-key features.

Instance-Key	Method	mIoU(T)	mIoU(O)
✘	mean discrepancy	37.8	38.8
	discrepancy similarity	39.1	40.2
✔	top-1 update	40.7	41.8
	top-50% update	42.2	44.3
	100% update	**44.1**	**46.9**

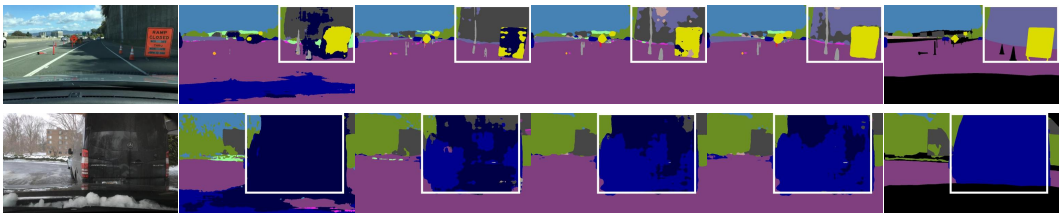

Figure 4: Segmentation results of different methods on the target domain of C-Driving.

In Table 4, we evaluate the performances of using discrepancy features in different ways. Removing all discrepancy features in OLDM, we again degrade the whole pipeline to the backbone segmentation network (see "w/o OLDM"). We also experiment with replacing discrepancy features with the discrepancy between the category- and instance-key features. This discrepancy only considers the difference between representative features of source and target domains. Our discrepancy features account for the differences between the representative features of source domain and various instances' object features of target domain. They show a stronger power for compensating the style information of relevant object instances. Moreover, we evaluate several alternatives for enabling discrepancy features for compensation. In contrast to our method that differentiates discrepancy features according to categories and instances, an alternative method merges all discrepancy features as a set (see "merged sets"). This method is insensitive to the specific properties of categories and instances. Another way of separating discrepancy features into multiple sets is to use k-means clustering (see "multi-sets, k-means"), without depending on category-key features. However, these methods neglect the useful category-key features, which select instance-key and discrepancy features for compensating the object features of the matched categories. Their performances lag behind our method (see "multi-sets, category").

Table 4: Results of various ways of using discrepancy features.

Discrepancy	Method	mIoU(T)	mIoU(O)
✘	w/o OLDM	36.6	39.7
	key discrepancy	42.7	44.3
✔	merged sets	39.6	41.5
	multi-sets, k-means	41.7	43.3
	multi-sets, category	**44.1**	**46.9**

Variants of Style Compensation In Tables 5 and 6, we evaluate the effectiveness of the style compensation by replacing it with other schemes during network testing. Here, OLDM has the optimized category-key, instance-key, and discrepancy features.

First, we remove the optimized OLDM and disable the style compensation. Another alternative method of style compensation is averaging all of the discrepancy features without relying on the similarities between target images' and instance-key features for weighting discrepancy features. The above methods (see "w/o OLDM" and "mean discrepancy") yield lower performances than our method, which better utilizes discrepancy features for compensating the style information of different categories and instances in target domain.

Table 5: Results of various compensation methods.

Style Compensation	Method	mIoU(T)	mIoU(O)
✘	w/o OLDM	36.6	39.7
✔	mean discrepancy	40.7	41.9
	instance similarity	**44.1**	**46.9**

Next, we analyze the quality of the pseudo annotations computed by the style compensation. Without pseudo annotations, we only use the ground-truth annotations of source images for training the segmentation network. We also evaluate the quality of the pseudo annotations produced by the intermediate segmentation head for supervising the segmentation network. Without the accurate pseudo annotations computed based on compensated object features, these alternatives yield worse results than our method.

Table 6: Results of various ways of computing pseudo annotations.

Pseudo Annotations	Method	mIoU(T)	mIoU(O)
✘	w/o pseudo	39.7	41.6
✔	intermediate	42.4	44.3
	final	**44.1**	**46.9**

5.3 External Comparison

In Table 7, we compare the performances of start-of-the-art OCDA , UDA and DG methods [1, 2, 3, 4, 5, 57]. For a fair comparison, the methods in each sub-table of Table 7 are trained on the same sets of images of source and target domains. They are tested on the same target and open domains. Our method of object style compensation outperforms an array of start-of-the-art methods on different datasets. In Figures 4 and 5, we compare the segmentation results of different methods, where our method yields better results.

Table 7: Comparison with state-of-the-art methods. We clarify the source, target, and open domains for training and testing the compared methods at the bottom of each sub-table. CD, CS, KT, and WD mean the mIoUs on the C-Driving, Cityscapes, KITTI, and WildDash datasets. mIoU^{11} and mIoU^{16} mean the mIoUs on 11 and 16 categories, respectively.

(a) Train: GTA5(Source), C-Driving(Target). Test: C-Driving(Target).

Method	Type	mIoU(T)
Source-only	-	28.3
CDAS[1]	OCDA	31.4
CSFU[3]	OCDA	34.9
DACS[57]	UDA	36.6
DHA[2]	OCDA	37.1
AST[4]	OCDA	38.8
ML-BPM[5]	OCDA	40.2
Ours	OCDA	**44.1**

(b) Train: SYNTHIA(Source), C-Driving(Target). Test: C-Driving(Target).

Method	Type	mIoU^{16}(T)	mIoU^{11}(T)
Source-only	-	20.9	28.1
CDAS[1]	OCDA	25.3	34.0
CSFU[3]	OCDA	26.1	34.8
DACS[57]	UDA	28.1	36.5
DHA[2]	OCDA	29.9	37.6
AST[4]	OCDA	31.1	38.9
ML-BPM[5]	OCDA	32.1	40.0
Ours	OCDA	**35.6**	**43.7**

(c) Train: GTA5(Source), C-Driving(Target). Test: C-Driving(Open), Cityscapes(Open), KITTI(Open), WildDash(Open).

Method	Type	CD	CS	KT	WD	Avg
CSFU[3]	OCDA	38.9	38.6	37.9	29.1	36.1
DACS[57]	UDA	39.7	37.0	40.2	30.7	36.9
RobustNet[58]	DG	38.1	38.3	40.5	30.8	37.0
DHA[2]	OCDA	39.4	38.8	40.1	30.9	37.5
AST[4]	OCDA	40.7	40.3	41.9	32.2	38.8
ML-BPM[5]	OCDA	42.5	41.7	44.3	34.6	40.8
Ours	OCDA	**46.9**	**43.6**	**46.5**	**40.1**	**44.3**

(d) Train: SYNTHIA(Source), C-Driving(Target). Test: C-Driving(Open), Cityscapes(Open), KITTI(Open), WildDash(Open).

Method	Type	CD	CS	KT	WD	Avg
CSFU[3]	OCDA	36.2	34.9	32.4	27.6	32.8
DACS[57]	UDA	36.8	37.0	37.4	28.8	35.0
RobustNet[58]	DG	37.1	38.3	40.1	29.6	36.3
DHA[2]	OCDA	38.9	38.0	40.6	30.0	36.9
AST[4]	OCDA	40.5	39.8	41.6	30.7	38.2
ML-BPM[5]	OCDA	42.6	41.1	43.4	30.9	39.5
Ours	OCDA	**48.5**	**48.0**	**51.3**	**39.6**	**46.9**

(e) Train: GTA5(Source), ACDC(Target). Test: ACDC(Target and Open).

Method	Type	mIoU(T)	mIoU(O)
Source-only	-	20.5	27.1
CDAS[1]	OCDA	25.3	29.1
CSFU[3]	OCDA	27.6	30.5
DACS[57]	UDA	29.0	34.8
DHA[2]	OCDA	29.5	37.5
AST[4]	OCDA	30.7	39.2
ML-BPM[5]	OCDA	32.1	41.6
Ours	OCDA	**35.7**	**44.1**

(f) Train: SYNTHIA(Source), ACDC(Target). Test: ACDC(Target and Open).

Method	Type	mIoU^{16}(T)	mIoU^{16}(O)
Source-only	-	19.8	20.5
CDAS[1]	OCDA	25.9	23.3
CSFU[3]	OCDA	26.7	24.8
DACS[57]	UDA	28.3	27.0
DHA[2]	OCDA	29.2	27.3
AST[4]	OCDA	30.1	27.9
ML-BPM[5]	OCDA	31.9	29.1
Ours	OCDA	**34.7**	**36.4**

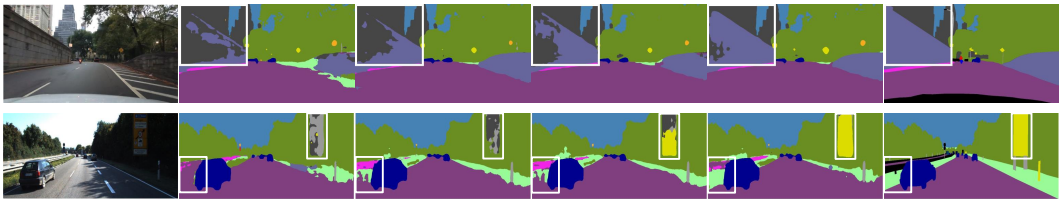

Figure 5: Segmentation results of various methods on the open domains of C-Driving and KITTI.

6 Conclusion

Open compound domain adaptation has been successfully used to improve semantic image segmentation performance. The popular methods globally adapt the scene styles of images. However, they unreasonably change the object styles of various categories and instances, forming unusual object contexts in the adapted images along with incorrect pseudo annotations. In this paper, we propose a novel idea of object-style compensation. Our method leverages the object-level discrepancy memory, where multiple sets of discrepancy features account for the style changes of objects in different categories. Furthermore, the discrepancy features in the same set individually consider instance-level style changes, thus adapting the object styles finely. Our method enables a more accurate computation of the pseudo annotations of the images in the target domains, which eventually assists in training the segmentation network. In the future, we will extend our method to other applications (e.g., object detection and image restoration).

References

[1] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X Yu, and Boqing Gong. Open compound domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12406–12415, 2020.

[2] Kwanyong Park, Sanghyun Woo, Inkyu Shin, and In So Kweon. Discover, hallucinate, and adapt: Open compound domain adaptation for semantic segmentation. *Advances in Neural Information Processing Systems*, 33:10869–10880, 2020.

[3] Rui Gong, Yuhua Chen, Danda Pani Paudel, Yawei Li, Ajad Chhatkuli, Wen Li, Dengxin Dai, and Luc Van Gool. Cluster, split, fuse, and update: Meta-learning for open compound domain adaptive semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8354, 2021.

[4] Jogendra Nath Kundu, Akshay R Kulkarni, Suvaansh Bhambri, Varun Jampani, and Venkatesh Babu Radhakrishnan. Amplitude spectrum transformation for open compound domain adaptive semantic segmentation. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 1220–1227, 2022.

[5] Fei Pan, Sungsu Hur, Seokju Lee, Junsik Kim, and In So Kweon. Ml-bpm: Multi-teacher learning with bidirectional photometric mixing for open compound domain adaptation in semantic segmentation. In *European Conference on Computer Vision*, pages 236–251, 2022.

[6] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021.

[7] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[8] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 126:961–972, 2018.

[9] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *European Conference on Computer Vision*, pages 402–416, 2018.

[10] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[11] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1094–1103, 2021.

[12] Ni Xiao and Lei Zhang. Dynamic weighted learning for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15242–15251, 2021.

[13] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *Advances in Neural Information Processing Systems*, 33:16282–16292, 2020.

[14] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13944, 2020.

[15] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12455–12464, 2020.

[16] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.

[17] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.

[18] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[19] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 749–757, 2020.

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

[21] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.

[22] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020.

[23] Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 9072–9081, 2021.

[24] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Curriculum graph co-teaching for multi-target domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5351–5360, 2021.

[25] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8187–8196, 2021.

[26] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *IEEE/CVF International Conference on Computer Vision*, pages 834–843, 2021.

[27] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578, 2020.

[28] Alexander Robey, George J Pappas, and Hamed Hassani. Model-based domain generalization. *Advances in Neural Information Processing Systems*, 34:20210–20229, 2021.

[29] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107, 2020.

[30] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[31] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 11749–11756, 2020.

[32] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

[33] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

[34] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6891–6902, 2021.

[35] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021.

[36] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021.

[37] Chaoqi Chen, Jiongcheng Li, Xiaoguang Han, Xiaoqing Liu, and Yizhou Yu. Compound domain generalization via meta-knowledge encoding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7129, 2022.

[38] Yan Huang and Liang Wang. Acmm: Aligned cross-modal memory for few-shot image and sentence matching. In *IEEE/CVF International Conference on Computer Vision*, pages 5774–5783, 2019.

[39] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020.

[40] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5802–5810, 2019.

[41] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[42] Huaibo Huang, Aijing Yu, and Ran He. Memory oriented transfer learning for semi-supervised image deraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7732–7741, 2021.

[43] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *IEEE/CVF International Conference on Computer Vision*, pages 13588–13597, 2021.

[44] Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12655–12663, 2020.

[45] Somi Jeong, Youngjung Kim, Eungbean Lee, and Kwanghoon Sohn. Memory-guided unsupervised image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6558–6567, 2021.

[46] Xin Feng, Wenjie Pei, Fengjun Li, Fanglin Chen, David Zhang, and Guangming Lu. Generative memory-guided semantic reasoning model for image inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7432–7447, 2022.

[47] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[48] Rui Xu, Minghao Guo, Jiaqi Wang, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Texture memory-augmented deep patch-based image inpainting. *IEEE Transactions on Image Processing*, 30:9112–9124, 2021.

[49] Tingliang Feng, Wei Feng, Weiqi Li, and Di Lin. Cross-image context for single image inpainting. *Advances in Neural Information Processing Systems*, 35:1474–1487, 2022.

[50] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021.

[51] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021.

[52] Vibashan Vs, Vikram Gupta, Poojan Oza, Vishwanath A Sindagi, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4516–4526, 2021.

[53] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[54] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4360, 2022.

[55] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118, 2016.

[56] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[57] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021.

[58] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021.