2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-4509-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CVPR46437.2021.00824

Cluster, Split, Fuse, and Update: Meta-Learning for Open Compound Domain Adaptive Semantic Segmentation

Rui Gong [1], Yuhua Chen [1], Danda Pani Paudel [1], Yawei Li [1], Ajad Chhatkuli [1], Wen Li [3], Dengxin Dai [1], Luc Van Gool [1,2]
[1] Computer Vision Lab, ETH Zurich, [2] VISICS, KU Leuven, [3] UESTC
{gongr, yuhua.chen, paudel, yawei.li, ajad.chhatkuli, dai, vangool}@vision.ee.ethz.ch,
liwenbnu@gmail.com

Abstract

Open compound domain adaptation (OCDA) is a domain adaptation setting, where target domain is modeled as a compound of multiple unknown homogeneous domains, which brings the advantage of improved generalization to unseen domains. In this work, we propose a principled meta-learning based approach to OCDA for semantic segmentation, MOCDA, by modeling the unlabeled target domain continuously. Our approach consists of four key steps. First, we ***cluster*** *target domain into multiple sub-target domains by image styles, extracted in an unsupervised manner. Then, different sub-target domains are* ***split*** *into independent branches, for which batch normalization parameters are learnt to treat them independently. A meta-learner is thereafter deployed to learn to* ***fuse*** *sub-target domain-specific predictions, conditioned upon the style code. Meanwhile, we learn to online* ***update*** *the model by model-agnostic meta-learning (MAML) algorithm, thus to further improve generalization. We validate the benefits of our approach by extensive experiments on synthetic-to-real knowledge transfer benchmark, where we achieve the state-of-the-art performance in both compound and open domains.*

1. Introduction

Semantic segmentation with minimal supervision is one of the most sought-after goals of image understanding [23, 56]. Unfortunately, the learned understanding in one domain does not generalize to the images from other domains [4]. In such cases, domain adaptation aims at transferring the shared knowledge across different but related domains [41], *i.e.,* source and target, using the unlabeled images from the target. When the target domain images are collected in mixed, continually varying, and even unseen conditions, understanding images invites the problem of open compound domain adaptation [35].

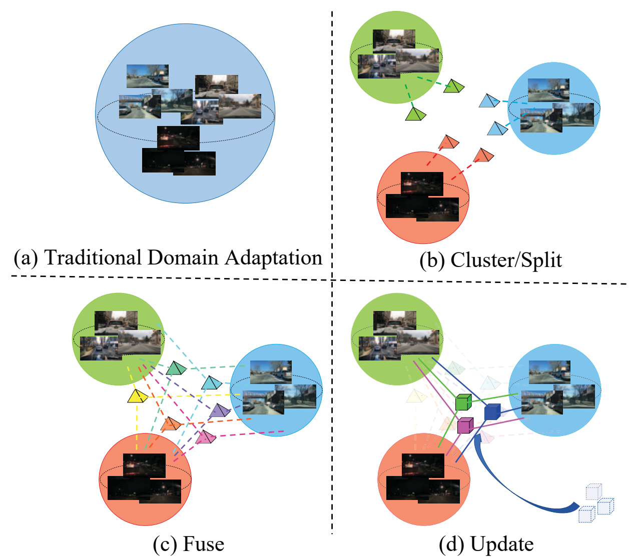


Figure 1: (a) The traditional unsupervised domain adaptation (UDA) vs. (b,c,d) the proposed meta-based open compound domain adaptation (MOCDA). Unlike the traditional UDA, MOCDA treats target as a compound of multiple unknown sub-domains. These sub-domains are discovered and processed using the cluster and the split module (b). The fuse module (c) then combines the sub-domain splits as basis (dash lines). On open domains, MOCDA adapts through online update during inference (blue arrow) in (d). Meta-learning serves in the fuse and the update module.

The Open Compound Domain Adaptation (OCDA) treats the target as a compound of multiple unknown sub-domains. Such assumption has been shown to be very promising by *Liu et al.* [35] for many practical settings of image classifications. However, the method developed in [35] does not fully exploit the same assumption for the task of image segmentation. [1] In this work, we show that the

[1]OCDA [35] does not fully exploit the domain information for seg-

978-1-6654-4509-2/21/$31.00 ©2021 IEEE
DOI 10.1109/CVPR46437.2021.00824

homogeneous sub-domain assumption can be exploited effectively also for image segmentation. We propose a novel meta-learning based approach to OCDA (abbreviated as MOCDA) that consists of four modules: cluster; split; fuse; and update, as illustrated in Fig. 1.

Similar to OCDA, the proposed MOCDA utilizes two image sets for training from: a single labeled source domain; and a diverse unlabeled target domain, which is assumed to be a compound of multiple unknown sub-domains. Such an assumption is suitable for real challenging situations, where the target domain is a combination of many factors including diverse weather, cities, and acquisition time [45, 13, 40]. The considered learning setup not only performs domain adaptation to the compound target domain, but also has generalization potential to unseen open domains. In this context, the process of domain adaptation happens to exhibit a meta-behaviour [29, 3, 12], which learned dynamically makes the open world semantic segmentation possible. In this work, we show that the meta-behaviour of OCDA can be learned using (a) a hypernetwork for dynamic fusion of knowledge, and (b) the online update. On the one hand, the update process – which is carried out using the model-agnostic meta-learning strategy – creates an opportunity for better open set generalization with only one gradient step. On the other hand, the learned dynamic fusion allows images to appear from the continuous manifold of the compound target domain.

In essence, the proposed framework serves in following four steps. (i) From target images, style codes are extracted and grouped into multiple clusters. (ii) For each cluster, a set of batch normalization (BN) parameters are learned. (iii) Corresponding to each cluster, each image can have different domain-specific predictions. The hypernetwork, then, learns to fuse these predictions. (iv) Model-agnostic meta-learning (MAML) [15] is exploited during hypertraining process, endowing the online update ability of the model on open domain during inference stage. The key contributions of this paper can be summarized as follows:

- We propose a novel framework for semantic segmentation in the OCDA setting. We use meta-learning in the dynamic fusion and MAML strategy based online update, to address the limitations of [35].

- We propose to model the compound target domain continuously, taking the sub-target domain as the basis, which offers the advantage of adapting to target domain and generalizing to unseen open domains.

- We demonstrate the adequacy of image style features, learned in an unsupervised manner, for our meta-based method MOCDA.

mentation task due to the inaccessibility of the domain encoder. Refer the original paper [35] for details.

- The proposed method provides the state-of-the-art results in synthetic-to-real knowledge transfer benchmark datasets, for both compound and open domains.

2. Related Works

Unsupervised Domain Adaptation and Generalization. Our work is related to domain adaptation [51, 41, 57, 18, 60, 44] and domain generalization [29, 28, 33, 30] works. Unsupervised domain adaptation aims at training a model on the labeled source domain and transferring the learned knowledge to the unlabeled target domain. The traditional unsupervised domain adaptation works [38, 16, 39, 59] typically focus on solving adaptation problem from a single source domain to a single target domain. Even though being effective in several tasks, the single target domain assumption is still restricted in many practical applications. Recently, multiple-target domain adaptation problem [12, 17] has received increasing research interests. The problem investigates knowledge transfer to multiple unlabeled target domains. Yet another important aspect not prioritized by the classical domain adaptation methods is the knowledge transfer to unseen but related open domains [35, 19, 31].

Cross-Domain Semantic Segmentation. In order to improve the adaptation and the generalization ability of the semantic segmentation model [7, 63, 50, 37, 9, 8], cross-domain semantic segmentation topic is extensively studied, both in the domain adaptation setting [68, 53, 71, 11, 10, 62] and in the domain generalization setting [61, 14, 19, 47, 35]. Most works either assume the target domain as a single domain [53, 68, 52, 22, 11, 10, 71], or a composition of multiple known domains [19, 67, 69, 47], with an exception of OCDA [35]. OCDA assumes target domain as a composition of multiple unknown domains, which is more realistic in practice. [35] follows a different approach for semantic segmentation compared to the classification task. The curriculum learning therefore is based on the average class confidence scores, rather than the neatly learned domain-focused factors in case of the classification task. Nevertheless, the experimental setup of our work is inspired by [35]. Concurrently, [43] develops the image translation based method for the OCDA problem, which is complementary to our method. Besides the open domain in [35, 43], our work further explores the generalization ability of the model when facing more diverse extended open domains.

Meta-Learning for Domain Adaptation/Generalization. Meta-learning addresses the problem of learning to learn and has been successfully applied to various applications including image classification [20], image restoration [24], visual tracking [5], and network compression [32]. The principle of meta-learning [54, 21] has also been investigated for the task domain adaptation [46, 27, 12] and generalization [29, 3, 14], with the algorithmic advances [2, 15, 48]. Our work can be related to those works in terms of general methodology. Among those works, the ones most related are [12] and [66]. The similarities are : 1) both of [12] and our MOCDA study the domain adaptation problem when there are multiple unknown target domains through meta-learning. 2) both of [66] and our MOCDA aims at improving the domain generalization performance for semantic segmentation model, with the help of MAML strategy. However, we have significant differences in the following aspects: 1) [12] utilizes the meta-learner for clustering the target domain into different sub-target domains, and the target domain is modeled as a union of multiple sub-target domains. And [12] does not include the open domain. However, our meta-hypernetwork is utilized to fuse the knowledge from different clusters, to model the target domain as a continuous compound target domain. 2) [66] does not study the domain adaptation problem, and only focus on the domain generalization. The MAML strategy in [66] is only used during training stage on the well labeled source domain. By contrast, MOCDA utilizes the MAML strategy in both of the well labeled source domain and the unlabeled target domain during the training stage. During inference, the MAML strategy is exploited for online update.

3. The MOCDA Model

Preliminaries. We consider that the labeled source domain $\mathcal{S}$ is composed of the source images $\mathbf{x}_s$, and the corresponding semantic labels $\mathbf{y}_s$, *i.e.*, $\mathcal{S} = \{(\mathbf{x}_s, \mathbf{y}_s) | \mathbf{x}_s \in \mathbb{R}^{H\times W\times 3}, \mathbf{y}_s \in \mathbb{R}^{H\times W}\}$, where H, W are height and width of the image, respectively. In OCDA, the unlabeled target domain $\mathcal{T}$ consists of target images $\mathbf{x}_t^i$ from multiple homogeneous sub-target domains, $\mathcal{T}^i = \{\mathbf{x}_t^i | \mathbf{x}_t^i \in \mathbb{R}^{H\times W\times 3}\}, i = 1, \ldots N$, where N is number of sub-target domains. In the context of this work (and also in OCDA), these sub-target domains are unknown. Therefore, the images $\mathbf{x}_t^i$ from some unknown sub-target domain $\mathcal{T}^i$ are simply denoted as $\mathbf{x}_t$, for notation convenience and clarity.

In this section, we propose the MOCDA model for semantic segmentation. The MOCDA model is composed of four modules: cluster, split, fuse, and update. The **Cluster** module extracts and clusters the style code from the target domain images automatically, dividing the target domain into multiple sub-target domains. The **Split** module adopts the compound-domain specific batch normalization (CDBN) layer to process different sub-target domain images using different branches. The **Fuse** module exploits a hypernetwork to predict the weights corresponding to each branch adaptively, conditioned on the style code of the input image. The final output of the network is the weighted combination of the outputs of different branches. The MAML method is utilized to train the **Fuse** module, so as to make the model be adapted quickly in **Update** module. Finally, the **Update** is carried out online during the inference time with one-gradient step, which is found to be beneficial for open domains. The framework overview is shown in Fig. 2a. In the following, we provide the details of all four modules, separately.

3.1. Cluster: Style Code Extraction and Clustering

The aim of the cluster module is to cluster the target domain $\mathcal{T}$ into different sub-target domains $\mathcal{T}^k, k = 1, \ldots, K$, serving the OCDA's assumptions of unknown multiple sub-target domains of the target domain. As shown in [35, 26], the major differences of the target domain images due to varying conditions, such as the weather, lighting, and inter-dataset, can be effectively reflected by the style of the images. Our cluster module consists of two mappings; $E_c(\cdot)$ and $E_l(\cdot)$. $E_c(\cdot)$ maps the target domain $\mathcal{T}$ to the style code domain $\mathcal{C}_t = \{\mathbf{c}_t | \mathbf{c}_t \in \mathbb{R}^l\}$ as $E_c : \mathcal{T} \to \mathcal{C}_t$, where l is the dimension of the style code. More specifically, the target domain image $\mathbf{x}_t$ is mapped to a low-dimension style code $\mathbf{c}_t = E_c(\mathbf{x}_t)$. Then a clustering algorithm, K-means [36], is adopted to automatically cluster the style code domain $\mathcal{C}_t$, partitioning into K clusters with centroids $\{\mathbf{c}_t^k\}$. We use the mapping $E_l(\cdot)$ to assign $\mathbf{x}_t$ to one of the sub-target domains, represented by the set $\mathcal{K} = \{k | k = 1, \ldots, K\}$, as $E_l : \mathcal{T} \to \mathcal{K}$. Here, we adopt the nearest neighbor strategy for $E_l(\cdot)$. More specifically, each target image is assigned to the nearest cluster, using the Euclidean distance between style codes of the image and the centroids, given by,

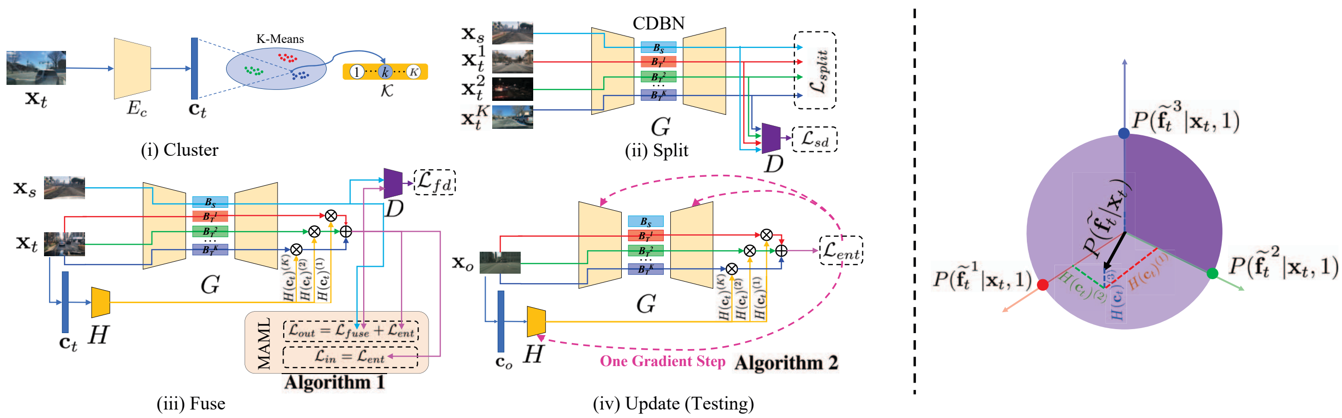
$$E_l(\mathbf{x}_t) := \arg\min_k \|\mathbf{c}_t - \mathbf{c}_t^k\|. \tag{1}$$

The key of our cluster module is to find an adequate mapping $E_c(\cdot)$. In this work, the unsupervised image translation framework MUNIT [26] is trained to translate between the source domain $\mathcal{S}$ and the target domain $\mathcal{T}$. During the translation training process, the style code encoder of MUNIT is trained to extract the style code from images unsupervisedly. The trained style encoder of MUNIT is used as $E_c(\cdot)$. Then, the target domain $\mathcal{T}$ is clustered into K sub-target domains $\mathcal{T}^k$, where the number of sub-target domains K is a hyperparameter. Using the nearest neighbour search, refer Eq. (1), each target image $\mathbf{x}_t$ is assigned to one of the sub-target domains $\mathcal{T}^k$. Henceforth, the image $\mathbf{x}_t$ assigned image k^{th} cluster is denoted as $\mathbf{x}_t^k$.

3.2. Split: Domain-Specific Batch Normalization

In [6], the domain-specific batch normalization (DSBN) is shown to be beneficial for the unsupervised domain adaptation (UDA), by separating the batch normalization layer for the source and target domain.

Similar to DSBN for UDA, the aim of our split module is to separate the multiple sub-target domain-specific information from the domain-invariant information. We propose DSBN for OCDA (abbreviated as CDBN), to conduct such separation for source domain $\mathcal{S}$ and the multiple



(a) Framework Overview (b) Compound Domain Modeling

Figure 2: (a) The overview of MOCDA framework demonstrating four modules; (i) Cluster, (ii) Split, (iii) Fuse, and (iv) Update. (b) Illustration of compound domain modeling, taking $K = 3$ for example. The sub-target domain $P(\widetilde{\mathbf{f}}_t^1|\mathbf{x}_t, 1)$, $P(\widetilde{\mathbf{f}}_t^2|\mathbf{x}_t, 2)$ and $P(\widetilde{\mathbf{f}}_t^3|\mathbf{x}_t, 3)$ is taken as basis. The cluster/split module models the compound target domain as the union set of three points, *i.e.,* red, green and blue points. But the fuse module models the compound target domain $P(\widetilde{\mathbf{f}}_t|\mathbf{x}_t)$ as the vector $H(\mathbf{c}_t) = [H(\mathbf{c}_t)^{(1)}, H(\mathbf{c}_t)^{(2)}, H(\mathbf{c}_t)^{(3)}]'$, composing the purple half quarter-spherical surface.

(clustered) sub-target domains $\{\mathcal{T}^k\}$. Note that DSBN for UDA learns only two sets of BN parameters (with possible extension given more labeled domains). However, the proposed CDBN learns $K+1$ sets of BN parameters for source domain and multiple *unlabeled* sub-target domains, *i.e.,* $B_S, B_T^1, ..., B_T^K$, formulated as,

$$B_S(\mathbf{x}_s, \mu_s, \sigma_s, \beta_s, \gamma_s) = \gamma_s \frac{\mathbf{x}_s - \mu_s}{\sigma_s} + \beta_s, \tag{2}$$

$$B_T^k(\mathbf{x}_t^k, \mu_t^k, \sigma_t^k, \beta_t^k, \gamma_t^k) = \gamma_t^k \frac{\mathbf{x}_t^k - \mu_t^k}{\sigma_t^k} + \beta_t^k, \tag{3}$$

where k is the sub-target domain label, $k = 1, ..., K$. Our split module replaces BN layers by CDBN. As shown in Fig. 2a, our split module includes the multi-branch semantic segmentation network $G = \{G_s, G_1, ..., G_K\}$ and the discriminator D. G_k is formed by selecting the k-th branch B_k of the CDBN layer. Through the adversarial learning, the discriminator D aligns the prediction distributions of source domain and that of the sub-target domains, in the output space. Therefore, the full optimization objective of the split module includes the semantic segmentation loss and the adversarial loss, presented below.

Semantic Segmentation Loss. We train the semantic segmentation network G with a standard cross entropy loss, using the source domain image $\mathbf{x}_s$ and the associated ground truth label $\mathbf{y}_s$,

$$\mathcal{L}_{seg}(G) = -\frac{1}{HW} \sum_{n=1}^{HW} \sum_{m=1}^{M} y_s^{(n,m)} \log(G_s(\mathbf{x}_s)^{(n,m)}), \tag{4}$$

where (n, m) represents (pixel, class) indices for M classes.

Multi-Branch Adversarial Loss. Recall the cluster module, each target image $\mathbf{x}_t$ is assigned to a unique sub-target domain label k, *i.e.,* $\mathbf{x}_t^k$. Here in the split module, the image $\mathbf{x}_t^k$ is processed using only the corresponding branch G_k, *i.e.,* $G_k(\mathbf{x}_t^k)$. Our multi-branch adversarial loss is an extension of the adversarial loss [58], which aligns the prediction distributions of the source domain $G_s(\mathbf{x}_s)$, and the sub-target domains $\{G_k(\mathbf{x}_t^k)\}$. The multi-branch adversarial loss $\mathcal{L}_{sadv}$ and the corresponding discriminator training loss $\mathcal{L}_{sd}$ are formulated as,

$$\mathcal{L}_{sadv}(G) = -\sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}_t^k \sim P_{T^k}} \log(D(G_k(\mathbf{x}_t^k))^{(n,1)}), \tag{5}$$

$$\begin{aligned} \mathcal{L}_{sd}(D) = & -\mathbb{E}_{\mathbf{x}_s \sim P_S} \log(D(G_s(\mathbf{x}_s))^{(n,1)}) \\ & -\sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}_t^k \sim P_{T^k}} \log(D(G_k(\mathbf{x}_t^k))^{(n,0)}), \end{aligned} \tag{6}$$

where P_S and P_{T^k} are the underlying data distributions of $\mathcal{S}$ and $\mathcal{T}_k$, respectively. The following full optimization objective is used for training our split module,

$$\mathcal{L}_{split}(G) = \mathcal{L}_{seg}(G) + \lambda_1 \mathcal{L}_{sadv}(G), \tag{7}$$

where λ_1 is a trades-off parameter. During the training process, we alternatively optimize the discriminator D and the generator G with the objective in the Eq. (6) and the Eq. (7), respectively.

3.3. Fuse: HyperNetwork for Branches Fusion

The **cluster** and **split** module discretizes the target domain into a few clusters, providing an initial discrete modeling of the target domain. The fuse of the discretized modes forms continuous manifold, the sample on which reflects the continuous change of the target domain and might correspond to an unseen domain. In the **fuse** module, we learn to combine the sub-target domain to model the compound target domain continuously.

Compound Domain Modelling. Here we model the target domain $\mathcal{T}$ in the corresponding feature domain $\mathcal{F}$, which is mapped by $F : \mathcal{T} \to \mathcal{F}$. Let $P(\widetilde{\mathbf{f}}_t^k|\mathbf{x}_t, k)$ be the feature distribution corresponding to image $\mathbf{x}_t$ when assumed to be from the k^{th} cluster. Then the distribution of the feature $\widetilde{\mathbf{f}}_t$ of the image $\mathbf{x}_t$, *i.e.,* $P(\widetilde{\mathbf{f}}_t|\mathbf{x}_t)$, is expressed as,

$$P(\widetilde{\mathbf{f}}_t|\mathbf{x}_t) = \sum_{k=1}^{K} P(\widetilde{\mathbf{f}}_t^k, k|\mathbf{x}_t) = \frac{1}{N}\sum_{k=1}^{K} P(k|\mathbf{x}_t)P(\widetilde{\mathbf{f}}_t^k|\mathbf{x}_t, k) \tag{8}$$

where $N = \int_{\widetilde{\mathbf{f}}_t^k} \sum_{k=1}^{K} P(\widetilde{\mathbf{f}}_t^k|\mathbf{x}_t, k)P(k|\mathbf{x}_t)\mathrm{d}\widetilde{\mathbf{f}}_t^k$. $P(k|\mathbf{x}_t)$ describes the probability distribution of the sub-target domain's label of image $\mathbf{x}_t$. By taking the sub-target domain distributions $P(\widetilde{\mathbf{f}}_t^k|\mathbf{x}_t, k)$ as basis, the compound target domain can be modeled with the vector, *i.e.,* $\{[P(1|\mathbf{x}_t), ..., P(k|\mathbf{x}_t), ..., P(K|\mathbf{x}_t)]'\}$.

HyperNetwork for Branches Fusion. In essence, the cluster and split module can be seen as modeling the sub-target domain label distribution as $P(k|\mathbf{x}_t) = 1, \text{if } E_l(\mathbf{x}_t) = k$ and $P(k|\mathbf{x}_t) = 0, \text{if } E_l(\mathbf{x}_t) \neq k$. It models the compound target domain as the discretized points in the vector space, as illustrated in Fig. 2b. In order to model the compound target domain in the continuous space, in our fuse module, we adopt the categorical distribution for $P(k|\mathbf{x}_t)$, *i.e.,*

$$P(k|\mathbf{x}_t) = w_k, \quad \text{with,} \quad \sum_{k=1}^{K} w_k = 1, w_k > 0, \tag{9}$$

where $\mathbf{w} = [w_1, ..., w_k, ..., w_K]^\top$ is the K-dimensional categorical vector, whose element w_k represents the probability that the target image $\mathbf{x}_t$ belongs to the sub-target domain $\mathcal{T}_k$. Then the hypernetwork $H(\cdot)$ is adopted to learn the $P(k|\mathbf{x}_t)$, by taking the style code $\mathbf{c}_t$ of the image sample $\mathbf{x}_t$ as input, *i.e.,* $[w_1, ..., w_k, ..., w_K]^\top = H(\mathbf{c}_t)$. Substituting the $H(\mathbf{c}_t)$ in Eq. (8), the feature distribution $P(\widetilde{\mathbf{f}}_t|\mathbf{x}_t)$ can be derived as,

$$P(\widetilde{\mathbf{f}}_t|\mathbf{x}_t) \sim \sum_{k=1}^{K} H(\mathbf{c}_t)^{(k)} P(\widetilde{\mathbf{f}}_t^k|\mathbf{x}_t, k). \tag{10}$$

where $H(\mathbf{c}_t)^{(k)}$ is the k^{th} element of $H(\mathbf{c}_t)$. Eq. (10) shows that the compound target domain is modeled in the continuous vector space, $H(\mathbf{c}_t)$, taking the sub-target domain distributions $P(\widetilde{\mathbf{f}}_t^k|\mathbf{x}_t, k)$ as basis, as illustrated in Fig. 2b.

From above, it is shown that $H(\mathbf{c}_t)$ weights the different sub-target domain distribution differently to get the compound target domain distribution. Here we adopt the network G as our mapping $\mathcal{F}$. Following [25], we reweight each feature sample $\widetilde{\mathbf{f}}_t^k = G_k(\mathbf{x}_t)$ with $H(\mathbf{c}_t)$, so that the feature sample from dominant sub-target domain has higher weight, whereas the sample from non-dominant sub-target domain has lower weight. The final prediction can be represented as,

$$\widetilde{\mathbf{y}}_t = \sum_{k=1}^{K} H(\mathbf{c}_t)^{(k)} G_k(\mathbf{x}_t). \tag{11}$$

By combining Eq. (11) and Eq. (5), the adversarial loss for the fuse module $\mathcal{L}_{fadv}$ and the corresponding discriminator training loss $\mathcal{L}_{fd}$ can be formulated as,

$$\mathcal{L}_{fadv}(G, H) = -\mathbb{E}_{\mathbf{x}_t \sim P_T} \log(D(\widetilde{\mathbf{y}}_t)^{(n,1)}) \tag{12}$$

$$\mathcal{L}_{fd}(D) = -\mathbb{E}_{\mathbf{x}_s \sim P_S} \log(D(G_s(\mathbf{x}_s))^{(n,1)}) - \mathbb{E}_{\mathbf{x}_t \sim P_T} \log(D(\widetilde{\mathbf{y}}_t)^{(n,0)}). \tag{13}$$

The optimization objective of our fuse module is a combination of Eq. (4) and Eq. (12), which is given by,

$$\mathcal{L}_{fuse}(G, H) = \mathcal{L}_{seg}(G) + \lambda_2 \mathcal{L}_{fadv}(G, H), \tag{14}$$

where λ_2 is the hyperparameter to balance between the adversarial loss and the segmentation loss. During the training process, we alternatively optimize the discriminator D and the generator G, the hypernetwork H with the objective in the Eq. (13) and the Eq. (14), respectively. In our MOCDA model, the training of the fuse module is combined with the MAML strategy, which is explained further in Section 3.4 and Algorithm 1.

3.4. Update: MAML based Online Update

In the previous OCDA work [35], the open set is only treated as a testing set to verify the generalization ability of the model. In contrast, in our work, the open set is also used for updating the model online during testing, for better generalization to the unseen domain, realized by MAML.

MAML. The MAML strategy [15] aims at learning the optimal model parameters θ^*, which eases the adaptation process for new tasks. In each iteration of MAML, there are two training loops; inner and outer. Let the data of inner and outer loops be $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$, respectively. In each training iteration, the model parameters θ are first updated with the inner loop loss $\mathcal{L}_{in}$ and data $\mathcal{D}_{in}$. The updated model is then evaluated on the outer loop loss $\mathcal{L}_{out}$ and data $\mathcal{D}_{out}$, to test the generalization ability of the updated model. Furthermore, the evaluation performance $\mathcal{L}_{out}$ is also adopted during update, to better generalize the model. This nested training fashion mimics the training and testing phase of the model. In order to endow adaptation ability, the optimization objective of MAML is formulated as,

$$\theta^* = \arg\min_{\theta} \ \mathcal{L}_{out}(\theta - \alpha\nabla\mathcal{L}_{in}(\theta, \ \mathcal{D}_{in}), \ \mathcal{D}_{out}), \tag{15}$$

where α is the learning rate for updating the model.

MAML for OCDA. In our addressed problem of OCDA for semantic segmentation, images from the set $\{\mathbf{x}_o\}$ of the unseen open domain $\mathcal{O}$ are available only during testing. We adopt the MAML algorithm in our MOCDA during training to be combined with the fuse module. MAML then offers us the advantage of quick adaptation to the open set during testing, by means of online update within one gradient step.

In the inner loop, we sample data from the target domain $\mathcal{T}$, *i.e.,* $\mathcal{D}_{in} = \{\mathbf{x}_t\}$. Meanwhile, in order to update the model without supervision, we use the unsupervised self-entropy loss[62] $\mathcal{L}_{ent}$ as the inner loop loss $\mathcal{L}_{in}$ – which mimics the model update process during testing, given by,

$$\mathcal{L}_{in} = \mathcal{L}_{ent} = -\frac{1}{HW}\sum_{n=1}^{HW}\sum_{c=1}^{C}\widetilde{\mathbf{y}}_t^{(n,c)}\log\widetilde{\mathbf{y}}_t^{(n,c)}. \quad (16)$$

In the outer loop, the data is sampled from both source domain $\mathcal{S}$ and the target domain $\mathcal{T}$, *i.e.,* $\mathcal{D}_{out} = \{\mathbf{x}_s, \mathbf{y}_s, \mathbf{x}_t\}$. In order to evaluate the model's performance on different domains and in different way, the outer loop loss $\mathcal{L}_{out}$ uses the optimization objective of the fuse module in Eq. (14) and the self-entropy loss in Eq. (16), such that,

$$\mathcal{L}_{out} = \mathcal{L}_{fuse} + \delta\mathcal{L}_{ent}, \quad (17)$$

where δ is the hyperparameter to balance between the fuse module loss and the unsupervised self-entropy loss. The MAML algorithm used during OCDA training is presented in Algorithm 1. Similarly, the MAML used during the online update, of OCDA testing, is given in Algorithm 2.

Algorithm 1 MAML algorithm for OCDA (Training)

Require: Source data $\mathcal{S} = \{(\mathbf{x}_s, \mathbf{y}_s)\}$, target data $\mathcal{T} = \{\mathbf{x}_t\}$, segmentation network G, hypernetwork H, discriminator D, the learning rate α of G, H, and the learning rate ζ of discriminator D.

1: Initialize the parameters θ_{GH} and θ_D, respectively of the segmentation network G, hypernetwork H, and the discriminator D;
2: **while** not done **do**
3: Sample $\mathcal{D}_{in}$ from $\mathcal{T}$ ▷ Inner Loop
4: $\theta_{GH}^{+} \leftarrow \theta_{GH} - \alpha\nabla_{\theta_{GH}}\mathcal{L}_{in}(\mathcal{D}_{in}, \theta_{GH})$;
5: Sample $\mathcal{D}_{out}$ from $\mathcal{S}$ and $\mathcal{T}$ ▷ Outer Loop
6: $\theta_{GH} \leftarrow \theta_{GH} - \alpha\nabla_{\theta_{GH}}\mathcal{L}_{out}(\mathcal{D}_{out}, \theta_{GH}^{+})$;
7: $\theta_D \leftarrow \theta_D - \zeta\nabla_{\theta_D}\mathcal{L}_{fd}(\mathcal{D}_{out}, \theta_D)$;
8: **end while**

Algorithm 2 MAML algorithm for OCDA (Testing)

Require: Data $\{\mathbf{x}_o\}$ from the unseen novel domain $\mathcal{O}$, segmentation network G, hypernetwork H.

1: Use trained parameters θ_{GH} of the segmentation network, G and the hypernetwork H, from the training phase;
2: $F \leftarrow 0$
3: **for** $i = 1, ..., n$ **do**
4: Sample the i^{th} image $\mathbf{x}_o^i$ from $\{\mathbf{x}_o\}$;
5: $\widetilde{\mathbf{y}_o^i} \leftarrow G(\mathbf{x}_o^i)$;
6: $\theta_{GH} \leftarrow \theta_{GH} - \eta\nabla_{\theta_{GH}}\mathcal{L}_{ent}(\widetilde{\mathbf{y}_o^i}, \theta_{GH})$
7: **end for**

3.5. Training Protocol of MOCDA

In total, our MOCDA model is trained in the multi-stage way, consisting of three steps: i) training the MUNIT model for style code extraction and clustering, ii) training with the CDBN layer in split module, iii) the CDBN layer is frozen, adding the hyper-network and the fuse module, and training the hypernetwork H and fine-tuning the semantic segmentation network G with MAML strategy as described in Algorithm 1. Then during testing stage, our whole model, except for CDBN layer, is online updated with the MAML strategy as clarified in Algorithm 2.

4. Experiments

In this section, we demonstrate the benefits of our MOCDA model under the open compound domain adaptive semantic segmentation setting. We compare our MOCDA model with other state-of-the-art (SOTA) methods on both of the target domain and the open domain. In order to further prove the effectiveness of our MOCDA model for open domain with online update, we introduce more diverse and challenging extended open domains to test the model performance additionally.

4.1. Experiments Setup

Following [35], we adopt the synthetic image dataset GTA5 [49] or SYNTHIA-SF [53] as the source domain, the rainy, snowy, and cloudy images in BDD100K[64] as the target domain, while the overcast images in BDD100K are utilized as the open domain. Besides, more diverse images from other real image datasets, Cityscapes[13], KITTI[1] and WildDash [65] are introduced as extended open domains. We adopt the the DeepLab-VGG16 model [7, 55] with the batch normalization layer as the segmentation network. The cluster numbers K is set as 4. The segmentation network and discriminator structure is the same as [58]. The hyperparameter λ_1 and λ_2 in Eq.(14) and Eq.(7) are set as 0.001. The hyperparameter δ in Eq.(17) is set as 0.0001.

4.2. GTA5 to BDD100K

Comparison with SOTA. In Table 1, we present our open compound domain adaptation results, in comparison with other SOTA methods. For fair comparison, all of the methods adopt the DeepLab-VGG16 model with the batch normalization layer. Compared with our baseline method AdaptSegNet[58], our split module achieves 3.1%

(a) Clustering visualization (b) Example images of different clusters

Figure 3: Visualization of clustering results. (a) is the t-SNE visualization of the style code extracted by the cluster module, (b) is example images from different clusters.

Source GTA→	Compound Rainy	Snowy	Cloudy	Open Overcast	Avg C	C+O
Source Only[35]	16.2	18.0	20.9	21.2	18.9	19.1
Source Only *	19.7	18.4	20.5	22.5	19.7	21.0
AdaptSegNet[35]	20.2	21.2	23.8	25.1	22.1	22.5
AdaptSegNet[58] *	21.6	20.5	23.9	27.1	22.3	24.4
CBST[71]	21.3	20.6	23.9	24.7	22.2	22.6
IBN-Net[42]	20.6	21.9	26.1	25.5	22.8	23.5
PyCDA [34]	21.7	22.3	25.9	25.4	23.3	23.8
OCDA [35]	22.0	22.9	27.0	27.9	24.5	25.0
Ours (Split)	23.5	23.5	27.8	29.5	25.4	27.1
Ours (Fuse)	**24.4**	**27.5**	**30.1**	**31.4**	**27.7**	**29.4**

Table 1: Semantic segmentation performance comparison with SOTA: GTA→ BDD100K with DeepLab-VGG16 backbone. The results are reported on mIoU over 19 classes. * means our reproduced result.

and 2.4% gain on the target domain and the open domain, respectively. Compared with the SOTA method OCDA[35], our split module performance outperforms by 0.9% on the target domain and by 1.6% on the open domain. It proves the effectiveness of our cluster module and the split module, for sub-target domain discovery and sub-target domain-specific information disjointing. The clustering visualization is shown in Fig. 3. Then by adopting the meta-learning with the hypernetwork and the MAML training strategy in the fuse module, our MOCDA model achieves the state-of-the-art performance, which improves the split module performance by 2.3% from 25.4% to 27.7%, and by 1.9% from 29.5% to 31.4% on the target domain and the open domain, respectively. It proves the advantage of our MOCDA model on fusing the different sub-target domains knowledge, modeling the target domain continuously through the hypernetwork, and adopting the MAML training strategy. The qualitative comparison of the semantic segmentation results on the target domain is shown in Fig. 4.

Online Update. Another meta-learning paradigm in our MOCDA model, besides the fuse module, is the MAML algorithm based online update during testing stage. From Table 2, it is shown that our MOCDA model without online update outperforms the baseline method AdaptSegNet [58] on both of the open domain and the extended open domain by 5.6% in average. It proves the effectiveness of our cluster, split and fuse module for open domain generalization. By further using the MAML based online update strategy described in Algorithm 2 during the testing

Source GTA→	Open BDD	Extended Open Cityscapes	KITTI	WildDash	Avg
Source[35]	21.2	–	–	–	–
Source*	22.5	19.3	24.1	16.0	20.5
AdaptSegNet[35]	25.1	–	–	–	–
AdaptSegNet[58] *	27.1	22.0	23.4	17.5	22.5
w/o Online Update	31.4	30.4	29.8	20.6	28.1
w/ Online Update	31.4	31.1	30.9	21.6	**28.8**
Gain of Online Update	–	+0.7	+1.1	+1.0	+0.7

Table 2: Open domain semantic segmentation performance comparison w/ or w/o online update: GTA→ BDD100K with DeepLab-VGG16 backbone. The results are reported on mIoU over 19 classes. * means our reproduced result.

stage, our MOCDA model performance on all the open domains improves by 0.7% in average, from 28.1% to 28.8%. Our model w/ or w/o online update has the same performance on the open domain, BDD100K overcast image. It is due to that the BDD100K overcast image is still from the BDD100K dataset, and the style gap between the overcast image and the target domain image is very narrow, whose visualization is shown in supplementary. The benefit from our cluster, split and fuse module has been able to handle the narrow style gap and have good generalization performance already. The performance gain, 0.7%, 1.1% and 1.0% on the extended open domains where the style gap is much larger, Cityscapes, KITTI and WildDash dataset, proves that the MAML based meta-learning paradigm, in Algorithm 1 for training and Algorithm 2 for testing, endows the fast adaptation ability to our model to generalize better on open domains. The qualitative comparison, w/ or w/o online update, on the open domains are shown in Fig. 4.

Ablation Study. We show the comparison of ablations and different variants of our model in Table 3. From Table 3, it is shown that all the modules, the cluster/split module ($\mathcal{L}_{split}$), the fuse module ($\mathcal{L}_{fadv}$) and the MAML training strategy are helpful to our whole MOCDA model. The cluster and split module has been proven to be helpful in the comparison with AdaptSegNet[58] and other SOTA methods. Here we show the effectiveness of our meta-learning paradigm, the hypernetwork and the MAML training strategy through the ablations and variants methods comparison. Firstly, in order to prove the validity of our hypernetwork, we build the baseline methods of the branch fusion in non-adaptive way; 1), averagely fuse for prediction during the testing stage of the split module. 2), averagely fuse during the training and testing stage of the fuse module. 3) use the style code distance from different clusters to weight different branches during the training and testing stage of the fuse module. It is shown that our hypernetwork based branch fusion strategy performance, 27.1%, outperforms all other non-adaptive fusion strategy, 23.1%, 26.1%, 26.6%. It benefits from the advantage of adaptive weights predicted from the hypernetwork conditioned on the image sample style code. Secondly, by comparing the performance of train-

$\mathcal{L}_{seg}$	$\mathcal{L}_{adv}$	$\mathcal{L}_{sadv}$	$\mathcal{L}_{fadv}$	$\mathcal{L}_{ent}$	MAML	mIoU
✓						18.9
✓	✓					22.3
✓		✓				25.4
✓		✓				23.1†
✓			✓			26.1‡
✓			✓			26.6§
✓			✓			27.1
✓			✓	✓		27.3
✓			✓	✓	✓	**27.7**

Table 3: Different ablations and variants comparison for OCDA, tested on BDD100k target domain based on DeepLab-VGG16 with batch normalization layer backbone. The results are reported on mIoU over 19 classes. † represents the average fusion only during testing. ‡ represents the average fusion of different branches during training and testing. § represents the style code distance weighted fusion during training and testing.

ing the fuse module using the $\mathcal{L}_{out}$ in the Eq. (17) and purely using the $\mathcal{L}_{fuse}$ in Eq.(14), it is shown that there is 0.2% performance gain by adding the unsupervised entropy loss, from 27.1% to 27.3%. By further introduce the MAML training strategy in Algorithm 1 for the fuse module, as done in our MOCDA model, the performance can be further improved to 27.7%. It proves that the MAML training strategy is not only helpful to the open domain generalization as described above, but also is beneficial to improve the adaptation performance of the model on the target domain. It results from that MAML training strategy mimics the training and testing procedure with the outer loop and inner loop and makes the model more domain adaptive.

4.3. SYNTHIA-SF to BDD100K

In this section, SYNTHIA-SF is used as the source domain. Following [70], we only take 11 main classes in the SYNTHIA-SF dataset to measure the semantic segmentation performance, which are road, sidewalk, building, wall, fence, pole, light, vegetation, sky, person and car.

Comparison with SOTA. In Table 4, we report the quantitative comparison results between our MOCDA model and other SOTA methods for the open compound domain adaptation setting, from the SYNTHIA-SF to the BDD100K. From Table 4, it is shown that our MOCDA model outperforms MinEnt [62] and AdaptSegNet [58] on both of the target domain and the open domain. It further verifies the effectiveness of our MOCDA model for OCDA.

Online Update. In Table 5, the performance of our MOCDA model for the open domain and the extended open domain are shown. Our MOCDA model w/o online update outperforms the AdaptSegNet method by 2.2% in average on all the open domains. By further utilizing the online update in the open domain, the peformance can be further improved by 1.1% in average, from 30.1% to 31.2%. It further proves the validity of the online update for the open domain.

Source SYNTHIA-SF→	Compound Rainy	Compound Snowy	Compound Cloudy	Open Overcast	Avg C	Avg C+O
Source Only	16.5	18.2	21.4	20.6	19.2	19.8
MinEnt[62]	21.8	22.6	26.2	25.7	23.9	24.7
AdaptSegNet[58]	24.9	26.9	30.7	30.3	28.0	29.0
Ours (Split)	25.2	27.9	32.4	31.8	29.1	30.3
Ours (Fuse)	**26.6**	**30.0**	**33.0**	**32.6**	**30.4**	**31.4**

Table 4: Semantic segmentation performance comparison with SOTA: SYNTHIA-SF→ BDD100K with DeepLab-VGG16 backbone. The results are reported on mIoU over 11 classes. The best results are denoted in bold.

Source SYNTHIA-SF→	Open BDD	Extended Open Cityscapes	Extended Open KITTI	Extended Open WildDash	Avg
Source	20.6	24.7	20.7	17.3	20.8
AdaptSegNet[58]	30.3	35.9	24.7	20.7	27.9
w/o Online Update	32.6	29.9	33.2	24.5	30.1
w/ Online Update	32.6	32.2	34.2	25.8	**31.2**
Gain of Online Update	–	+2.3	+1.0	+1.3	+1.1

Table 5: Open domain semantic segmentation performance comparison w/ or w/o online update: SYNTHIA-SF→ BDD100K with DeepLab-VGG16 backbone. The results are reported on mIoU over 11 classes.

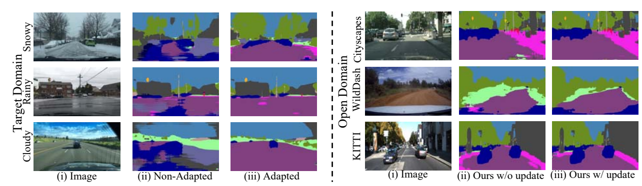


Figure 4: Qualitative comparison of semantic segmentation results on the target domain, including the rainy, snowy and cloudy weather, and on the open domains, KITTI, WildDash and Cityscapes.

5. Conclusion

In this paper, we address the problem of open compound domain adaptation, and propose a meta-learning based model, MOCDA. MOCDA is composed of four modules, cluster, split, fuse and update module. Meta-learning serves in fuse and update module for continuously modeling the compound target domain and online update. The extensive experiments show that our model achieves the state-of-the-art performance on different benchmarks, proving the effectiveness of our proposed MOCDA model.

Acknowledgements. This research has received funding from the EU Horizon 2020 research and innovation programme under grant agreement No. 820434. Wen Li is supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400. Dengxin Dai is supported by the Toyota Motor Europe via the research project TRACE Zurich.

References

[1] 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. *IJCV*, 126(9):961–972, 2018. 6

[2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016. 2

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018. 2

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 1

[5] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 2

[6] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, 2019. 3

[7] 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. *TPAMI*, 40(4):834–848, 2017. 2, 6

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[9] 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. 2

[10] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *CVPR*, 2019. 2

[11] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, 2018. 2

[12] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *CVPR*, 2019. 2, 3

[13] 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. 2, 6

[14] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 2019. 2

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2, 5

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 2

[17] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *TIP*, 29:3993–4002, 2020. 2

[18] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 2

[19] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019. 2

[20] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2

[21] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *ICANN*, 2001. 2

[22] 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. 2

[23] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 1

[24] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A magnification-arbitrary network for super-resolution. In *CVPR*, 2019. 2

[25] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *NeurIPS*, 2007. 5

[26] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 3

[27] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. *arXiv preprint arXiv:2004.04398*, 2020. 2

[28] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 2

[29] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017. 2

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

[31] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. Domain generalization and adaptation using low rank exemplar svms. *TPAMI*, 40(5):1114–1127, 2017. 2

[32] Yawei Li, Shuhang Gu, Kai Zhang, Luc Van Gool, and Radu Timofte. DHP: Differentiable meta pruning via hypernetworks. *arXiv preprint arXiv:2003.13683*, 2020. 2

[33] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018. 2

[34] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-

domain semantic segmentation: A non-adversarial approach. In *ICCV*, 2019. 7

[35] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X Yu, and Boqing Gong. Open compound domain adaptation. In *CVPR*, 2020. 1, 2, 3, 5, 6, 7

[36] Stuart Lloyd. Least squares quantization in pcm. *TIT*, 28(2):129–137, 1982. 3

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[38] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 2

[39] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017. 2

[40] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *IJRR*, 36(1):3–15, 2017. 2

[41] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *TNN*, 22(2):199–210, 2010. 1, 2

[42] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 7

[43] Kwanyong Park, Sanghyun Woo, Inkyu Shin, and In-So Kweon. Discover, hallucinate, and adapt: Open compound domain adaptation for semantic segmentation. In *NeurIPS*, 2020. 2

[44] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2

[45] Matthew Pitropov, Danson Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Waslander. Canadian adverse driving conditions dataset. *arXiv preprint arXiv:2001.10117*, 2020. 2

[46] Kun Qian and Zhou Yu. Domain adaptive dialog generation via meta learning. *arXiv preprint arXiv:1906.03520*, 2019. 2

[47] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, 2020. 2

[48] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019. 2

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

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[51] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2

[52] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 2

[53] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. *arXiv preprint arXiv:1711.06969*, 2017. 2, 6

[54] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn*. PhD thesis, Technische Universität München, 1987. 2

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[56] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 1

[57] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2

[58] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 4, 6, 7, 8

[59] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 2

[60] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 2

[61] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, 2018. 2

[62] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 2, 6, 8

[63] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. *arXiv preprint arXiv:2101.11939*, 2021. 2

[64] 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

[65] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *ECCV*, 2018. 6

[66] Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Generalizable semantic segmentation via model-agnostic learning and target-specific normalization. *arXiv preprint arXiv:2003.12296*, 2020. 3

[67] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *NeurIPS*, 2019. 2

[68] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017. 2

[69] 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 *NeurIPS*, 2019. 2

[70] Brady Zhou, Nimit Kalra, and Philipp Krähenbühl. Domain adaptation through task distillation. In *ECCV*, 2020. 8

[71] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 2, 7