Source-Free Open Compound Domain Adaptation in Semantic Segmentation

Yuyang Zhao*, Zhun Zhong*,†, Zhiming Luo, Gim Hee Lee, Nicu Sebe

***Abstract*—In this work, we introduce a new concept, named source-free open compound domain adaptation (SF-OCDA), and study it in semantic segmentation. SF-OCDA is more challenging than the traditional domain adaptation but it is more practical. It jointly considers (1) the issues of data privacy and data storage and (2) the scenario of multiple target domains and unseen open domains. In SF-OCDA, only the source pre-trained model and the target data are available to learn the target model. The model is evaluated on the samples from the target and unseen open domains. To solve this problem, we present an effective framework by separating the training process into two stages: (1) pre-training a generalized source model and (2) adapting a target model with self-supervised learning. In our framework, we propose the Cross-Patch Style Swap (CPSS) to diversify samples with various patch styles in the feature-level, which can benefit the training of both stages. First, CPSS can significantly improve the generalization ability of the source model, providing more accurate pseudo-labels for the latter stage. Second, CPSS can reduce the influence of noisy pseudo-labels and also avoid the model overfitting to the target domain during self-supervised learning, consistently boosting the performance on the target and open domains. Experiments demonstrate that our method produces state-of-the-art results on the C-Driving dataset. Furthermore, our model also achieves the leading performance on CityScapes for domain generalization.**

***Index Terms*—Semantic Segmentation, Open Compound Domain Adaptation, Source-free Domain Adaptation.**

I. Introduction

DEEP learning has now achieved a remarkable success in fully-supervised semantic segmentation [1]–[6], which, however, is relied heavily on the expensive dense pixel-wise annotations. One solution to lighten the labeling cost is unsupervised domain adaptation (UDA), which aims to transfer the knowledge of labeled synthetic data to unlabeled real-world data. Despite the effectiveness of existing UDA methods [7]–[9], they mainly consider the context of a single target domain, resulting in limited applications in the real world. Indeed, the target domain may be captured from multiple data distributions without a clear separation and the system will unavoidably face instances from unseen domains. To investigate a more realistic domain adaptation problem, in this paper, we consider the setting of open compound domain adaptation (OCDA) [10] for semantic segmentation. In OCDA, the unlabeled target domain is a compound of multiple homogeneous domains without domain labels. The adapted model is applied to test samples from the compound target domain and an open domain, where the open domain is unseen during training.

TABLE I
COMPARISONS OF DIFFERENT CROSS-DOMAIN TRANSFER LEARNING SETTINGS. **DA**: DOMAIN ADAPTATION, **SF-DA**: SOURCE-FREE DA, **DG**: DOMAIN GENERALIZATION, **OCDA**: OPEN COMPOUND DA, **SF-OCDA**: SOURCE-FREE OCDA.

Settings	Source Data	Source Model	Unlabeled Target	Multiple Targets	Open Targets
DA [7]	✓	✓	✓	✗	✗
SF-DA [11]	✗	✓	✓	✗	✗
DG [12]	✗	✓	✗	✗	✓
OCDA [10]	✓	✓	✓	✓	✓
SF-OCDA	✗	✓	✓	✓	✓

Existing UDA [7], [8], [13], [14] and OCDA [10], [15], [16] methods commonly require the use of the labeled source data during the whole training process. However, the source data are not always available due to data privacy. In addition, the source data are generally very large, which require plenty of storage space (*e.g.*, GTA5 [17]≈57GB). This further limits the applications of existing methods, especially when transferring to a lightweight self-driving device. Nevertheless, we can choose to maintain the pre-trained source model instead of the source data, enabling us to obey the data privacy policy and use much less storage space (*e.g.*, DeepLab-VGG16 [2], [18]≈120MB). These facts motivate us to introduce a more challenging but practical setting for OCDA, called source-free OCDA (**SF-OCDA**), where only the source pre-trained model and the unlabeled target data are available during the training of the target model. In the literature, source-free domain adaptation (SF-DA) has recently been developed in image classification [19], [20] and semantic segmentation [11], [21] for the single target case. However, as shown in Tab. I and Fig. 1, compared with SF-DA, our SF-OCDA demands not only adapting to data from multiple target domains but also considering the generalization performance on unseen domains.

In SF-OCDA, the source data and target data are invisible

* Equal contribution. † Corresponding author.

This work is supported by the EU H2020 project AI4Media No. 951911, the PRIN project CREATIVE Prot. 2020ZSL9F9, the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG2-RP-2020-016), and the Tier 2 grant MOE-T2EP20120-0011 from the Singapore Ministry of Education.

Y. Zhao and G. Lee are with the Department of Computer Science, National Univeristy of Singapore, 117417, Singapore. (Email: yuyang.zhao@u.nus.edu, gimhee.lee@nus.edu.sg)

Z. Zhong and N. Sebe are with the Department of Information Engineering and Computer Science (DISI), University of Trento, Trento, TN 38122, Italy. (Email: zhun.zhong@unitn.it, sebe@disi.unitn.it)

Z. Luo is with the Department of Artificial Intelligence, Xiamen University, Xiamen, 361005, China. (Email: zhiming.luo@xmu.edu.cn)

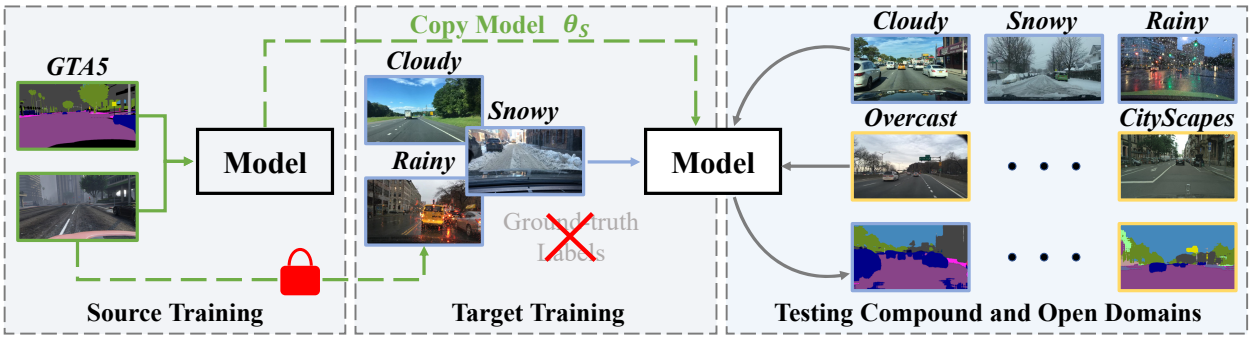


Fig. 1. Illustration of source-free open compound domain adaptation (SF-OCDA). In the training stage, the model is first trained on the (synthetic) labeled source data and then adapted to the (real-world) unlabeled compound target data. The source data are **not available** during the target adaptation. In the testing stage, the learned model is used to predict the semantic segmentation results for samples from the compound and open domains.

to each other. In such context, we cannot align the domain distributions as traditional UDA methods [7]–[9], since they require the co-occurrence of source and target data. Moreover, existing UDA methods only focus on the performance on the target domain at hand while ignoring the performance on open unseen domains. Instead, this paper introduces an effective two-stage framework for SF-OCDA, which consists of (1) training a generalized source model and (2) adapting the target model with self-supervised learning. In the first stage, we aim to learn a robust model, which can generalize well to different target domains. To achieve this goal, we propose the **C**ross-**P**atch **S**tyle **S**wap (CPSS), which can effectively augment the samples with various image styles. Specifically, CPSS first extracts the styles of patches in feature maps and then randomly exchanges the styles among patches by the instance normalization and de-normalization. In this manner, CPSS can prevent the model from overfitting to the source domain and thus significantly improve the generalization ability of the model. In the second stage, we adapt the target model by self-supervised learning. Specifically, we optimize the target model with the guide of pseudo-labels generated from the pre-trained source model, which can implicitly align the source and target distributions under the constraint of label consistency. Moreover, CPSS is also applied to reduce the influence of noisy pseudo-labels and to avoid overfitting to the target domain, which can further boost the performance on the compound and open domains. Our contributions are summarized as follows:

- We introduce a new setting for semantic segmentation, *i.e.*, source-free open compound domain adaptation (SF-OCDA), which is an important yet unstudied problem. In addition, we propose an effective framework for solving SF-OCDA, which focuses on learning a generalized model during the stages of source pre-training and target adaptation.
- We propose the CPSS, which diversifies the samples in the feature-level, to improve the generalization ability of the model in both source and target training stages. CPSS is a lightweight module without learnable parameters, which can be readily injected into existing segmentation models.
- The proposed framework learned with the source-free constraint significantly outperforms the state-of-the-art methods on the OCDA benchmark. Our approach also surpasses the advanced domain generalization approaches on CityScapes.

II. Related Work

Semantic Segmentation. With the development of deep neural networks (DNNs), fully-supervised methods [1]–[6], [22] have achieve significant performance. FCN [1] first introduces the fully convolutional networks to solve the pixel-wise segmentation in an end-to-end manner. DeepLab [2] improves the performance by enlarging the receptive field with atrous convolutional layers. Recently, CTNet [22] models the long-term dependency by leveraging spatial and channel contextual information to further improve the semantic representation. To alleviate the heavy annotation cost, weakly-supervised semantic segmentation [23]–[25] flourishes in the community. The mainstream of weakly-supervised setting leverages the image-level labels to train the segmentation model with the help of CAM [26]. Change *et al.* [24] investigate the object sub-categories to improve the completeness of CAMs. SSA [23] explores the multi-stage semantic structure information to refine the high-quality CAMs. Both fully-supervised and weakly-supervised settings focus on the closed-set semantic segmentation, while there also exist cases that the segmentation classes change. Continual semantic segmentation [27]–[29] is introduced to address the existence of new classes. Furthermore, it is possible that the segmentation system faces the class change when applying to new scenes, where both the domain shift and category shift impair the model performance. To solve this problem, continual domain adaptation [30], [31] is proposed to narrow the domain shift between multiple domains while dealing with new classes without forgetting. Different from previous works, we focus on open compound domain adaptation in semantic segmentation in this paper. We aim at addressing the domain shift between the source domain and the complex target domains, including the unlabeled compound domain and unseen open domain.

Transfer Learning in Semantic Segmentation. To tackle the expensive cost of collecting and labeling real-world data, transfer learning has attracted a widespread attention in semantic segmentation. Commonly, transfer learning methods are developed along two directions: unsupervised domain adaptation (UDA) and domain generalization (DG). UDA aims

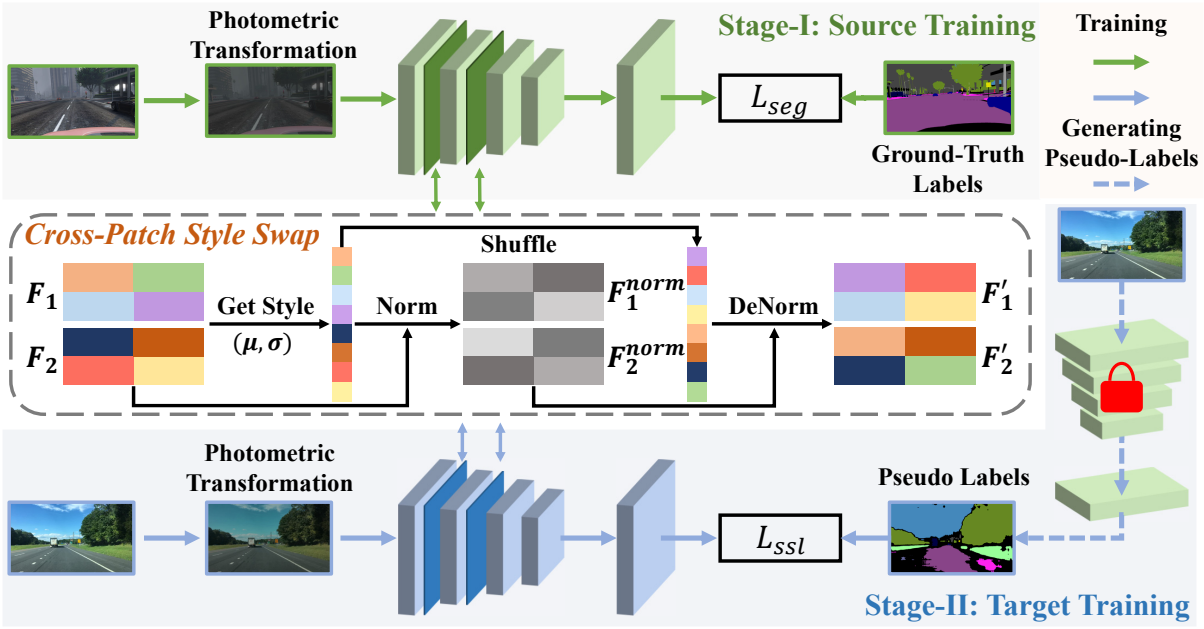


Fig. 2. The framework of the proposed method. (1) The model is first trained on the labeled source domain. (2) We generate pseudo-labels by the source pre-trained model and train the target model in a self-training manner. In the second stage, we have no access to the source data. To improve the generalization ability of the model, we equip the model with the Cross-Patch Style Swap module in the two training stages, which augments features by exchanging styles among patches.

at transferring the knowledge from a labeled source domain to an unlabeled target domain. Existing UDA approaches can be roughly divided into two categories, *i.e.*, aligning domain distributions through adversarial learning [7], [13], [32]–[34] and self-training on the target domain [8], [9], [35]–[40]. Kang *et al.* [34] build the pixel-level cycle association between source and target pixel pairs and enhance the connections contrastively for narrowing the domain gap. Zheng and Yang [9] utilizes uncertainty estimation to refine target pseudo-labels for learning target domain. MRNet [36] adopts a two-stage strategy, using adversarial learning of dual classifier in the first stage and leveraging memory regularization to refine the self-training in the second stage. DG focuses on training a robust model with synthetic data, which can generalize well on unseen real-world target data. To reduce the large gap between synthetic data and the real-world data, DG methods usually augment the synthetic samples [12], [41] with the styles of ImageNet [42] or conditionally align the outputs [43], [44] between the segmentation model and the ImageNet pre-trained model. In addition, some works are proposed to learn domain-invariant features by removing domain-specific information [45], [46] or feature augmentation [47]. Recently, Liu *et al.* [10] propose the setting of open compound domain adaptation (OCDA), which can be regarded as an extension of UDA and DG. In OCDA, the model trained with source and target data is used to evaluate samples from the compound target domain and unseen open domain. Liu *et al.* [10] introduce a memory-based curriculum learning framework to improve the generalization on the compound and open domains. [15] and [16] discover the latent target domains, aligning the source and latent domains with multiple domain discriminators. Different from these OCDA methods, this work investigates the OCDA under the source-free constraint and aims to learn a robust model by augmenting features with patch styles.

Source-Free Domain Adaptation. Hypothesis transfer learning (HTL) [48] aims to retain the prior knowledge in a form of hypothesis instead of data for the source domain. However, the main drawback of HTL is that it requires a small set of labeled target data. Inspired by HTL, source-free domain adaptation (SF-DA) [11], [19]–[21], [49]–[53] has recently flourished. In SF-DA, in contrast to the source data, the source pre-trained model is provided in the target training stage. SHOT [19] maintains the source hypothesis by fixing the trained classifier and maximizes mutual information of target outputs for distribution alignment. Kundu *et al.* [20] generate negative samples by image composition, which are used to narrow domain shift and category gap during source training. In addition, an instance-level weighting mechanism is proposed for effective target adaptation. Lately, Liu *et al.* [11] introduce source-free domain adaptation for semantic segmentation and utilize the batch normalization statistics of the source model to recover source-like samples. MAS^3 [54] learns prototypical distribution from the source domain for aligning the distributions across domains in the embedding space under the source-free constraint. Kundu *et al.* [55] develop the multi-head framework to improve the generalization ability with the virtually extended multi-source dataset and extract reliable target pseudo-labels for self-training. In this work, we introduce the source-free open compound domain adaptation (SF-OCDA) for semantic segmentation, extending SF-DA to a more realistic setting. MRNet [36] and Kundu *et al.* [55] also adopt two-stage training strategy and investigate domain adaptation in semantic segmentation. Nevertheless, the main difference between our method and the above two methods lies in the setting. First, both MRNet [36] and Kundu *et al.* [55] focus on the single target domain adaptation,

(a) Source Domain (GTA5) (b) Target Domain (C-Driving)

Fig. 3. Visualization of style distributions for (a) source domain and (b) target domain. We use the concatenation of mean and standard deviation of the feature map after the first block of VGG16 [18] as the style feature and show the 2D embeddings by t-SNE [56]. Zoom in for details.

where only the performance on one unlabeled target domain is evaluated. Our SF-OCDA setting focuses on both the unlabeled compound domain and the unseen open domain, which is more challenging and practical than [36] and [55] . Second, despite using the two-stage training strategy, MRNet [36] is not under the source-free constraint, where both labeled source domain and unlabeled target domain is used simultaneously in the first stage. The comparison between SF-OCDA and existing adaptation settings is reported in Tab. I.

III. Methodology

Preliminaries. In open compound domain adaptation (OCDA) [10], we are given a labeled (synthetic) source domain $\mathcal{S}$ and an unlabeled (real) compound target domain $\mathcal{T}$. The goal is to train a model that can accurately predict semantic labels for instances from the compound and open target domains. Specifically, $\mathcal{S}$ includes N_S images $x_i^s \in \mathbb{R}^{H^s \times W^s \times 3}$ and their corresponding semantic labels $y_i^s \in \mathbb{R}^{H^s \times W^s}$ of C classes. $\mathcal{T}$ contains N_T images $x_i^t \in \mathbb{R}^{H^t \times W^t \times 3}$ of multiple homogeneous domains without semantic and domain labels. In this paper, **we consider the setting of source-free OCDA (SF-OCDA)**, which imposes an extra constraint that only the pre-trained source model, instead of the source data, is available for training the target model together with the unlabeled target data.

A. Overview

In this section, we propose an effective framework (shown in Fig. 2) for SF-OCDA, which separates the training process into two stages: (1) training a generalized source model and (2) adapting a target model with self-supervised learning. We also introduce the Cross-Patch Style Swap (CPSS) to augment features with various patch styles, which can significantly improve the generalization ability of the model in both training stages. The motivation for our two-stage pipeline is as follows. To meet the source-free constraint, the two-stage learning pipeline is essential, *i.e.*, learning a source model and then adapting a target model. In such a context, we are motivated to learn robust models in each stage. Therefore, we propose to learn a source model that can generalize well to unseen potential real-world data, which can largely benefit the subsequent target adaptation stage. Another way to address the absence of source data is to learn a image generation model in the source training stage, which can recover the source data or source distribution in the target adaptation stage. However, learning an effective image generation model is very difficult especially for the complex driving scenes. In addition, quality of the recovered images should be very high since the the segmentation task is the pixel-level classification task. Therefore, learning a generalized source model is a more feasible and effective strategy. In the second stage, we suggest to narrow the domain gap between the source and target data with self-training, as the source data are not accessible. Note that, domain alignment of existing domain adaptation methods can not be readily applied due to the source-free constraint. In this stage, pseudo-labels play an important role and we focus on resisting the impact of noisy pseudo-labels. To this end, we propose the CPSS that can serve both stages but plays different roles. In the first stage, CPSS mainly aims to avoid the model overfitting to the source data, improving the generalization ability. In the second stage, CPSS not only helps the model generalize to unseen open domains, but also aims to reduce the impact of noisy pseudo-labels, leading to a more reliable self-training.

Next, we first introduce our CPSS module (Sec. III-B) and then present the proposed training strategy (Sec. III-C) in detail.

B. Cross-Patch Style Swap

Motivation. Image style variation is an important factor that influences the model performance in semantic segmentation. Although the synthetic data are generated to simulate the real-world images, the styles of synthetic images are still very different from those of the real ones. Therefore, the model trained on the synthetic data is sensitive to the real style variations and thus produces poor performance on real images. To this end, we attempt to learn a robust model, which is insensitive to style variations, by augmenting the training samples with diverse styles.

In order to implement style augmentation, the key is extracting style factors from images. To achieve this goal, we draw inspiration from style transfer [57], [58], which obtains image styles by extracting the mean μ and standard deviation σ of the feature map in neural networks. In Fig. 3, we visually verify the feasibility of using the μ and σ as the style features

in semantic segmentation. It is clear that images of various styles (*e.g.*, day and night) can be well-separated by the style features. In addition, AdaIN [59] shows that an input sample can be transferred to an arbitrary style while retaining the semantic content, by replacing the style features. AdaIN is formulated as:

$$\text{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y), \tag{1}$$

where $\mu(.)$ and $\sigma(.)$ denote the mean and standard deviation of the input feature map, respectively. x and y are two feature maps that provide the semantic content and the image style, respectively. Inspired by AdaIN, we propose two style augmentation operations based on the style features of image patches for training a robust segmentation model.

Intra-Image Cross-Patch Style Swap. In the self-driving scenario, different patches (*e.g.*, up and down) of a frame may include different objects, such as sky, vehicle, road and fence, making these patches present different styles. Intuitively, we can generate a new stylized sample by exchanging the style features of different patches. Hence, we propose the intra-image Cross-Patch Style Swap. Specifically, the feature map of an image is first separated into $n = n_h \times n_w$ patches:

$$F = \begin{bmatrix} F_{1,1} & \cdots & F_{1,n_w} \\ \vdots & \ddots & \vdots \\ F_{n_h,1} & \cdots & F_{n_h,n_w} \end{bmatrix}. \tag{2}$$

Then, each patch is normalized by the mean and standard deviation of itself, and de-normalized by the style feature of a random patch, formulated by:

$$F^{'}_{i,j} = \sigma(\tilde{F}_{i,j})\left(\frac{F_{i,j} - \mu(F_{i,j})}{\sigma(F_{i,j})}\right) + \mu(\tilde{F}_{i,j}), \tag{3}$$

where $F^{'}_{i,j}$ denotes the style swapped counterpart of $F_{i,j}$. $\tilde{F}_{i,j}$ denotes the shuffled patch that provides the style feature.

Inter-Image Cross-Patch Style Swap. Although the intra-image CPSS can enrich the styles of a feature map, the model can easily remember the intra-image style variations after several training epochs, which will limit the effectiveness of the CPSS. However, the patch styles vary greatly among different images, which can be used to further enhance the style diversity during CPSS. Taking this into consideration, we introduce the inter-image CPSS, which collects style features from all the patches in a mini-batch with B samples and exchanges these styles ($B \times n$) among all patches. We re-formulate Eq 3 as:

$$F^{'}_{k,i,j} = \sigma(\tilde{F}_{k,i,j})\left(\frac{F_{k,i,j} - \mu(F_{k,i,j})}{\sigma(F_{k,i,j})}\right) + \mu(\tilde{F}_{k,i,j}), \tag{4}$$

where $F^{'}_{k,i,j}$ denotes the swapped counterpart of patch $F_{i,j}$ in the kth sample. $\tilde{F}_{k,i,j}$ denotes a randomly selected patch that provides the style feature.

CPSS is injected into several layers of the backbone, which is activated in the training stage with a probability of β and is not used in the testing stage.

Photometric Transformation. In practice, the brightness, contrast and saturation of the frame vary in different situations. For example, images are brighter in the sunny morning while the contrast is stronger in snowy weather. In addition, there may exist blurry images caused by the rainy weather. Consequently, we randomly apply photometric transformation to the input images, including color jitter, Gaussian blur and grayscale, to simulate the real-world style various, which can further improve the effect of CPSS.

C. Model Training

As shown in Fig. 2, our framework includes two stages, *i.e.*, the source training stage and the target training stage, where the source data and target data are used independently in their own stages.

Stage-I: Source Training. In this stage, we aim at training a generalized model with synthetic labeled source domain $\mathcal{S}$. We adopt the cross-entropy loss to train the model, formulated as:

$$L_{seg} = -\frac{1}{HW}\sum_{m=1}^{HW}\sum_{c=1}^{C} y^{s}_{m,c} \log p^{s}_{m,c}, \tag{5}$$

where $y^{s}_{m,c}$ denotes the ground truth for the mth pixel and $p^{s}_{m,c}$ denotes the softmax probability of this pixel belonging to the cth class. Importantly, we employ the proposed CPSS along with photometric transformation to augment the samples in both feature- and image-levels, which can effectively improve the generalization ability of the source model.

Stage-II: Target Training. For SF-OCDA, source data are not available in this stage. Instead, we are given the source pre-trained model and the unlabeled compound target domain $\mathcal{T}$ to learn a target model that can perform well on both compound and open domains. In this stage, the target model is cloned from the source pre-trained model and trained in a self-supervised manner.

Specifically, we first generate pseudo-labels based on the predictions of the source pre-trained model by maximum probability threshold (MPT) [60]. MPT estimates class thresholds based on the top $q\%$ pixels of each class and a predefined threshold τ. The pseudo-labels are then assigned to pixels where the prediction values of the dominant classes are higher than the corresponding class thresholds.

With the pseudo-labels, we employ the cross-entropy loss to enforce the consistency between the outputs of source and target models:

$$L_{ssl} = -\frac{1}{HW}\sum_{m=1}^{HW}\sum_{c=1}^{C} \hat{y}^{t}_{m,c} \log p^{t}_{m,c}, \tag{6}$$

where $p^{t}_{m,c}$ is the prediction of the target model and $\hat{y}^{t}_{m,c}$ is the generated pseudo-label. Note that, we only update the model with high confident pixels that are assigned with pseudo-labels, which tend to be less noisy.

Similar to Stage-I, we also adopt CPSS and the photometric transformation to train the target model, which brings two advantages. First, due to the unsatisfactory performance of the source training model, the target pseudo-labels are inevitably noisy. If we directly utilize the original samples for self-training, the model will overfit to the noisy labels after training for some iterations. However, if we diversify the target samples by CPSS and PT while using the pseudo-labels from original samples, the model can focus more on the consistency among

TABLE II
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON GTA5 → C-DRIVING. † DENOTES METHODS THAT EMPLOY THE LONG-TRAINING STRATEGY.

Methods GTA5 →	Backbone	Source Free	Compound(C) Rainy	Compound(C) Snowy	Compound(C) Cloudy	Open(O) Overcast	Avg C	Avg C+O
Source Only	VGG16	✓	16.2	18.0	20.9	21.2	18.9	19.1
AdaptSeg [7]		✗	20.2	21.2	23.8	25.1	22.1	22.5
CBST [35]		✗	21.3	20.6	23.9	24.7	22.2	22.6
IBN-Net [46]		✗	20.6	21.9	26.1	25.5	22.8	23.5
PyCDA [65]		✗	21.7	22.3	25.9	25.4	23.3	23.8
Liu *et al.* [10]		✗	22.0	22.9	27.0	27.9	24.5	25.0
Park *et al.* [15]		✗	27.0	26.3	30.7	32.8	28.5	29.2
Source Only†	VGG16	✓	23.6	24.4	27.8	29.5	25.6	26.3
AdaptSeg [7]†		✗	25.6	27.2	31.8	32.1	28.8	29.2
MOCDA [16]†		✗	24.4	27.5	30.1	31.4	27.7	29.4
Park *et al.* [15]†		✗	27.1	30.4	35.5	36.1	32.0	32.3
Ours (Stage-I)†		✓	28.5	30.5	36.4	37.4	32.8	33.2
Ours (Stage-II)†		✓	**30.6**	**31.9**	**37.6**	**38.0**	**34.4**	**34.5**
Source Only†	ResNet101	✓	27.6	27.8	32.9	33.0	30.0	30.3
Ours (Stage-I)†		✓	**35.5**	33.4	41.4	41.2	37.8	37.9
Ours (Stage-II)†		✓	35.3	**36.9**	**41.8**	**42.0**	**38.5**	**39.0**

different styles than fitting to the pseudo-labels. The similar phenomenon is also observed in semi-supervised learning [61], [62], which utilize strong-augmented samples for self-training with noisy pseudo-labels. Second, the model requires to perform on unseen open domains, *e.g.*, overcast [10], [63] and CityScapes [64], which are of different styles from the unlabeled compound domain, As stated in Stage-I, one role of CPSS is to avoid the model overfitting to the training data by diversifying samples. Therefore, CPSS is also used to improve the generalization ability on unseen open domains. These two advantages improve the model performance on the target compound and open domains.

IV. EXPERIMENTS

A. Experimental Setup

Datasets. Following [10], [15], we use the synthetic image data GTA5 [17] and SYNTHIA [66] as the source domain, the rainy, snowy, and cloudy images in C-Driving [10], [63] as the compound target domain, and the overcast images in C-Driving as the open domain. To further measure the generalization ability of models, we additionally use Cityscapes [64] as an extended open domain. GTA5 includes 24,966 training images with a resolution of 1914×1052, and SYNTHIA [66] contains 9,400 images of 960×720. C-Driving consists of 14,697 unlabeled training images and 1,430 testing images, where the image size is 1280×720. Cityscapes contains 500 images of 2048×1024 for validation.

Evaluation Metric. We use mean intersection-over-union (mIoU) to evaluate the semantic segmentation performance. For GTA5 → C-Driving, we use the 19 shared semantic categories for training and evaluation. When using SYNTHIA as the source domain, we use 16 shared categories, ignoring the train, truck, and terrain categories.

Implementation Details. We use the DeepLab-V2 [2] with VGG16 [18] and ResNet101 [67] backbone as the segmentation model. For the source training stage, following [7], [15], we use SGD with an initial learning rate 2.5×10^{-4}, momentum 0.9 and weight decay 5×10^{-4} to optimize the model. For the target training stage, the learning rate is reduced to 1×10^{-4}. In both stages, we use the polynomial decay with a power of 0.9 to schedule the learning rate. The total training process takes 150K iterations, with a batch size of 1. We set τ=0.9 and q%=50% for generating pseudo-labels. For CPSS, the number of patches n and the activation probability β are set to 4 and 0.3, respectively. By default, we use the inter-image CPSS and inject it after the first and second blocks of the backbone. Note that, we use 4 samples for CPSS, but optimize the model with only the first image. This can greatly reduce the computational cost since using a batch size of 1 or 4 achieves a similar performance. The overall training time for the two stages is 24 GPU hours on one NVIDIA GTX 2080Ti GPU.

B. Comparison with State-of-the-Art Methods

Results of GTA5 → C-Driving. In Tab. II, we compare our method with the state-of-the-art UDA models [7], [35], [46], [65] and OCDA models [10], [15], [16] on the setting of "GTA5 → C-Driving". For a fair comparison, all the models adopt DeepLab-V2 with VGG16 backbone. Following [15], we use the long training scheme (150K iterations) to train the model. We make the following observations. First, the models trained with the long training scheme produce higher results, showing the advantage of the long training scheme. Second, our Stage-I model, which is trained only with the source data, achieves the best performance among all the existing methods that use both the source and the target data. This verifies the effectiveness of the proposed CPSS in learning a generalizable model. Third, our Stage-II model outperforms all compared models by a large margin, indicating that our method produces new state-of-the-art performance for OCDA, even under the source-free constraint. In addition, we also provide the results of our method with ResNet101 backbone. As shown in Tab. II, our Stage-I model outperforms the baseline model by 7.8% in

TABLE III
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON SYNTHIA → C-DRIVING. ALL MODELS ARE TRAINED WITH THE LONG TRAINING STRATEGY, USING VGG16 BACKBONE. WE REPORT AVERAGED PERFORMANCE ON 16 CLASS SUBSETS FOLLOWING THE EVALUATION PROTOCOL USED IN [13], [15]. * DENOTES THE SOURCE ONLY MODEL TRAINED IN THIS PAPER.

Methods SYNTHIA →	Source Free	Compound(C) Rainy	Snowy	Cloudy	Open(O) Overcast	Avg C	C+O
Source Only [15]	✓	16.3	18.8	19.4	19.5	18.4	18.5
CBST [35]	✗	16.2	19.6	20.1	20.3	18.9	19.1
CRST [68]	✗	16.3	19.9	20.3	20.5	19.1	19.3
AdaptSeg [7]	✗	17.0	20.5	21.6	21.6	20.0	20.2
Advent [13]	✗	17.7	19.9	20.2	20.5	19.3	19.6
Park *et al.* [15]	✗	18.8	21.2	23.6	23.6	21.5	21.8
Source Only*	✓	18.9	19.7	20.4	21.3	19.7	20.1
Ours (Stage-I)	✓	**22.4**	23.8	**25.3**	**26.4**	24.0	24.5
Ours (Stage-II)	✓	**22.4**	**24.5**	**25.3**	**26.4**	**24.2**	**24.7**

C mIoU and 7.6% in C+O mIoU, and our target training stage yields 1.1% improvement in C+O mIoU.

Results of SYNTHIA → C-Driving. In Tab. III, we compare our method with state-of-the-art methods [7], [13], [15], [35], [68] on the setting of "SYNTHIA → C-Driving". All models adopt VGG16 backbone. Clearly, (1) the proposed method largely improves the performance of the source only model, and (2) our two models (Stage-I and Stage-II) both significantly outperform the state-of-the-art methods, verifying the generalization ability of the proposed method with different source datasets. We also find that the improvement of our Stage-II is limited. This is because given a poorly trained source model (≈ 24% mIoU), we fail to generate enough useful / reliable pseudo-labels for self-supervised learning on the target domain. In our experiments, training the target model without the proposed CPSS will reduce the performance. This phenomenon can also be observed for Advent [13], which additional uses entropy information to train the AdaptSeg [7] but achieves lower results on C-Driving (in Tab. III). In contrast, using our CPSS can alleviate the impact of wrong pseudo-labels and can guarantee that self-supervised learning will not hamper the model performance.

Results of Domain Generalization. We also verify the generalization ability of our method on CityScapes in Tab. IV. We can observe that our Stage-I model surpasses the state-of-the-art domain generalization methods with both VGG16 and ResNet101 backbone when trained only with GTA5. Compared with DRPC [12] that additionally uses ImageNet [42] images, our model outperforms it by 1.0% and 2.2% in mIoU with VGG16 and ResNet101 backbone respectively. These findings demonstrate the effectiveness of the proposed method on open domains.

TABLE IV
EVALUATION ON OPEN DOMAIN CITYSCAPES. § EXTRA USING THE IMAGENET IMAGES.

Method	GTA5 → CityScapes VGG16	ResNet101
ASG [43]	31.5	32.8
IBN-Net [46]	34.8	40.3
DRPC [12]§	36.1	42.5
Ours (Stage-I)	**37.1**	**44.7**

TABLE V
EFFECTIVENESS OF STYLE AUGMENTATIONS.

Model	CPSS	PT	C	C+O
Stage-I	✗	✗	25.6	26.3
	✗	✓	27.1	27.4
	✓	✗	31.2	32.0
	✓	✓	**32.8**	**33.2**
Stage-II	✗	✗	33.3	33.5
	✗	✓	33.7	33.5
	✓	✗	34.3	34.4
	✓	✓	**34.4**	**34.5**

TABLE VI
COMPARISON OF DIFFERENT STYLIZED OPERATIONS.

Method	C	C+O
MixStyle [69]	30.7	31.2
CrossNorm [47]	31.4	31.8
CPSS (intra-image)	31.7	32.3
CPSS (inter-image)	**32.8**	**33.2**

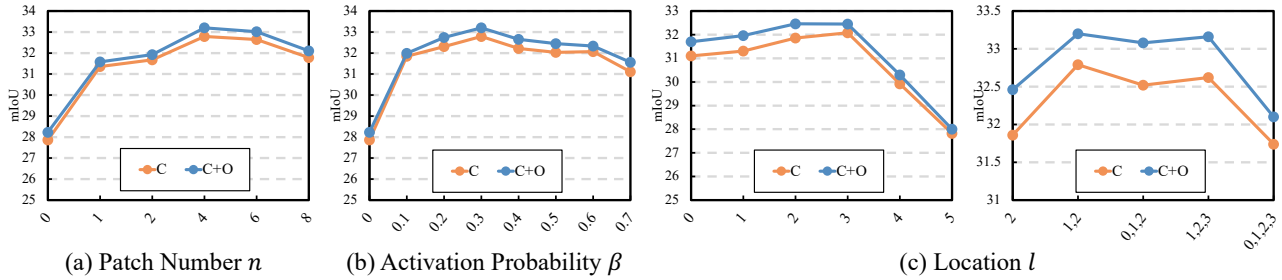
TABLE VII
IMPACT OF LATENT DOMAINS.

W/ Latent	Split	C	C+O
✓	Clustering	**34.4**	**34.7**
	Oracle	34.3	34.5
✗	—	**34.4**	34.5

C. Evaluation

In this section, we evaluate the effectiveness and superiority of the proposed method. Experiments are conducted with VGG16 backbone on the setting of "GTA5 → C-Driving".

Effectiveness of Style Augmentations. In Tab. V, we investigate the effectiveness of the proposed CPSS and photometric transformation (PT). Clearly, CPSS consistently improves the performance for both stages. Specifically, for the source training stage (Stage-I), inserting CPSS outperforms the baseline by 5.6% in C mIoU and by 5.7% in C+O mIoU. Adopting PT only can also improve the performance for both compound and open domains, yielding 1.5% and 1.1% improvement in



(a) Patch Number n (b) Activation Probability β (c) Location l

Fig. 4. Sensitivities to (a) the number of patches, (b) activation probability and (c) injecting location of CPSS in the source training stage.

TABLE VIII
INFLUENCE OF THE NUMBER OF SAMPLES s FOR CPSS.

s	1	2	4	8	16
C	31.7	32.2	**32.8**	32.5	32.7
C+O	32.3	32.9	33.2	33.4	**33.5**

C mIoU and C+O mIoU. Adopting the photometric transformation on top of CPSS further gains 1.6% and 1.2% improvement in C mIoU and C+O mIoU, respectively. Both PT and CPSS diversify the limited source data with different styles and variations, so the model can be less overfitting to the synthetic source domain but generalize better to the unseen real-world target domain. For the target training stage (Stage-II), we initialize the model by the source model trained with CPSS and PT. It should be notice that, if we directly use the source model without augmentation to adapt target model, the performance will degrade due to the poor performance of the source model. This further underlines the importance of learning a generalized source model in SF-OCDA. In Stage-II, self-supervised learning achieves limited improvement without using style augmentations. In contrast, adding CPSS can clearly promote performance on both compound and open domains. This verifies that CPSS can not only reduce the impact of noisy samples but also improve the robustness of the model to unseen domains. On the other hand, using PT can hardly gain improvements in the target adaptation. This is mainly because the model has been familiar with such transformation during source training. The above results also indicate the effectiveness of our two-stage pipeline. In the source training stage, the model is first trained with diverse source samples, which can generalize well to the unseen target domain. Then we fine-tune the source pre-trained model with unlabeled target data, so the model can better fit to real-world data. With the two-stage pipeline, the model can perform well on both the compound and unseen real-world data.

Comparison of Different Stylized Operations. The proposed CPSS is closely related to MixStyle [69] and CrossNorm [47], which are both designed for domain generalization. All three methods aim to improve the generalization ability of the model by perturbing style features of training samples. However, the stylized operations of them are different. Specifically, MixStyle replaces the style of a sample with the one that is generated by mixing its own style feature with a shuffled style feature using a random convex weight. Instead, CrossNorm directly exchanges the styles of two samples, which is a special case of MixStyle when the weight of the shuffled style feature is 1. Both MixStyle and CrossNorm compute one style feature for each sample and stylize each sample with one style feature. Different from them, our CPSS generates several styles for each sample by separating the feature map into multiple patches. This modification is specially designed for semantic segmentation in the self-driving scenario since patches in a frame could contain different styles. Compared with MixStyle and CrossNorm, our CPSS can provide more diverse and useful styles for generating stylized feature maps. In addition, with CPSS, the model is trained with richer feature maps where each one contains multiple different styles, further enforcing the model to be robust to style variations. In Tab. VI, we compare MixStyle [69], CrossNorm [47], and two versions of our CPSS. Experiments are conducted in the source training stage. We can find that mixing styles with a random weight (MixStyle) is less suitable for semantic segmentation, because MixStyle may sometimes generate semantically unrealistic styles. Compared with CrossNorm and CPSS (intra-image), CPSS (inter-image) produces clearly higher performance. This indicates that augmenting samples with more various styles can help us to learn a more generalizable model and our CPSS can better diversify existing styles.

Is Splitting Latent Domains Necessary? Recent OCDA methods [15], [16] show that the sub-domain labels can be used to reduce the latent domain gaps in the target domain. Instead, in our target training stage, we randomly select training samples from the target data to form the mini-batch without considering the sub-domain labels. To verify the impact of considering the latent domains for CPSS, we implement our framework with a new sampling strategy. Specifically, we sample the images in a balanced way, so that each mini-batch contains at least one sample for each sub-domain. We provide two kinds of latent domains: "Oracle" denotes using the original rainy, snowy, cloudy as the latent domains; and "Clustering" denotes separating latent domains by clustering the style features. As shown in Tab. VII, the random sampling strategy and its two variants achieve similar performance. This indicates that the proposed CPSS can potentially consider the style variations among multiple latent domains and learn a robust model.

D. Parameter Analysis

We further analyze the sensitivities of CPSS to four important hyper-parameters, *i.e.*, the activation probability β, the

TABLE IX
EFFECTIVENESS OF CPSS IN ADAPTSEG MODEL FOR OCDA (GTA5→C-DRIVING) AND UDA (GTA5→CITYSCAPES). ALL MODELS USE VGG16 BACKBONE. * DENOTES REPRODUCING THE METHOD BASED ON THE SOURCE CODE.

Methods	CPSS	GTA5→C-Driving						GTA5→CityScapes
		Compound(C)			Open(O)	Avg		
		Rainy	Snowy	Cloudy	Overcast	C	C+O	
AdaptSeg [7]	✗	—	—	—	—	—	—	35.0
AdaptSeg [7]*	✗	25.6	27.2	31.8	32.1	28.8	29.2	34.2
AdaptSeg [7]	✓	**28.9**	**29.1**	**35.2**	**36.0**	**31.9**	**32.3**	**38.5**

TABLE X
EFFECTIVENESS OF CPSS FOR SOURCE-FREE DOMAIN ADAPTATION (GTA5 → CITYSCAPES).

Method	GTA5 → CityScapes	
	ResNet50	ResNet101
Source Only	34.0	35.9
SFDA [11]	43.2	—
Sivaparased [21]	—	45.1
Ours (Stage-I)	42.2	44.7
Ours (Stage-II)	**45.0**	**47.2**

number of patches n, the injecting location l and the number of samples s for CPSS. Experiments are conducted in the source training stage with VGG16 backbone.

Patch Number n. We compared the results of using different numbers of patches n in Fig. 4(a). The model is trained without CPSS when n=0. With the increase of n, the model is encouraged to face more styles, producing higher results. However, the performance is degraded when n is large, *i.e.*, 8. Moreover, the model fails to converge when continuing to increase n (e.g., 64 / 128). This is mainly because the extracted style features may contain excessive semantic information instead of the style information when patches are quite small, which will impair the semantic representation of each patch during style exchanging.

Activation Probability β. In Fig. 4(b), we investigate the effect of the probability β of activating the CPSS operation. The performance first increases with the value of β and peaks when β=0.3. However, assigning a larger value to β (*e.g.*, 0.7) leads to performance degradation. The results show that diversified styles can improve the generalization but training with excessive generated styles fails to further improve the model performance.

Injecting Location l. In Fig. 4(c), we investigate the impact of injecting CPSS into different blocks of the network. Block-0 denotes injecting CPSS before the network (image-level), and block-l ($l > 0$) denotes injecting CPSS before the last pooling layer of the lth convolutional block. We make two observations. First, injecting CPSS into shallow layers, *i.e.*, block-$0, 1, 2, 3$, helps to improve the performance, while the performance degrades when injecting CPSS into a deep layer (block-$4, 5$). The reason is that the mean and standard deviation represent style information in shallow layers but contain more semantic information in deep layers. Second, jointly injecting into multiple (two or three) layers can achieve further improvement. Considering the trade-off between accuracy and runtime, injecting CPSS into block-1 and block-2 is an appropriate choice.

Number of samples s for CPSS. We investigate the influence of the number of samples s for CPSS in Tab. VIII. Inter-image CPSS will degenerate to intra-image CPSS when s is set to 1. As shown in Tab. VIII, the performance of compound domain first improves with the increase of s and peaks when $s = 4$. On the other hand, the overall performance of compound and open domains consistently improves with the increase of s but the gain is very limited when $s = 16$. The results show that enlarging the number of CPSS samples can generate more diversified samples and thus can improve the generalization ability. However, the improvement of enlarging s is quite slight after enough candidate styles are provided since each image is diversified by a specific number of styles (4 patch for each image in this paper). Consequently, considering the trade-off between the computational cost and performance, we set s to 4 in our experiments.

E. Additional Experimental Results

Performance on unsupervised domain adaptation. To further demonstrate the generalization ability of the proposed CPSS, we conduct experiments on unsupervised domain adaptation (UDA). We inject CPSS into the widely used domain adaptation approach, AdaptSeg [7], and evaluate the results on the settings of "GTA5 → C-Driving" and "GTA5 → CityScapes" in Tab. IX. Note that, when using AdaptSeg, the source-free constraint is not enforced. Clearly, CPSS can consistently improve the performance of AdaptSeg by a large margin on both settings. This further confirms the compatibility of the proposed CPSS.

Performance on source-free domain adaptation. Source-free domain adaptation (SF-DA) is the UDA setting under source-free constraint. Instead of multiple compound and open domains in SF-OCDA, the target domain of SF-DA is only a single domain, which is easier but less practical than SF-OCDA. In Tab. X, we compare our method with two recently proposed SF-DA methods [11], [21] in the GTA5 → CityScapes setting. Note that [11] is evaluated on ResNet50 while [21] is evaluated on ResNet101. As shown in Tab. X, our approach outperforms [11] and [21] by 1.8% and 2.1%, respectively. Such findings demonstrate the advantage of the proposed method in the SF-DA setting.

Per-Class IoU on GTA5 → C-Driving. In Tab. XI, we report the per-class IoU on different sub-domains of "GTA5→C-Driving". Generally, our methods (Stage-I and Stage-II) produce higher results on most classes for all sub-domains. On the other hand, we find that all the methods fail to recognize the

TABLE XI
PER-CLASS IOU ON DIFFERENT SUB-DOMAINS OF THE OCDA BENCHMARK: GTA5 → C-DRIVING. THE RAINY, SNOWY AND CLOUDY WEATHER COMPOSE THE COMPOUND TARGET DOMAIN, WHILE THE OVERCAST WEATHER IS THE OPEN DOMAIN. THE RESULTS ARE REPORTED OVER 19 CLASSES. THE "BICYCLE" CLASS IS NOT LISTED DUE TO THE RESULT IS CLOSE TO ZERO. THE BEST RESULTS ARE DENOTED IN BOLD. † DENOTES METHODS THAT EMPLOY THE LONG-TRAINING STRATEGY.

		GTA5→C-Driving																		
Sub-domain	Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motocycle	mIoU
Rainy	Source Only [10]	48.3	3.4	39.7	0.6	12.2	10.1	5.6	5.1	44.3	17.4	65.4	12.1	0.4	34.5	7.2	0.1	0.0	0.5	16.2
	AdaptSegNet [7], [10]	58.6	17.8	46.4	2.1	19.6	15.6	5.0	7.7	55.6	**20.7**	65.9	17.3	0.0	41.3	7.4	3.1	0.0	0.0	20.2
	CBST [10], [35]	59.4	13.2	47.2	2.4	12.1	14.1	3.5	8.6	53.8	13.1	80.3	13.7	**17.2**	49.9	8.9	0.0	0.0	**6.6**	21.3
	IBN-Net [10], [46]	58.1	19.5	51.0	4.3	16.9	18.8	4.6	9.2	44.5	11.0	69.9	20.0	0.0	39.9	8.4	15.3	0.0	0.0	20.6
	OCDA [10]	63.0	15.4	54.2	2.5	16.1	16.0	5.6	5.2	54.1	14.9	75.2	18.5	0.0	43.2	9.4	24.6	0.0	0.0	22.0
	MOCDA [16]†	66.8	22.0	52.4	6.7	16.7	16.9	5.3	3.5	60.4	17.2	80.1	21.8	0.1	46.4	17.9	**29.4**	0.0	0.0	24.4
	Source Only†	65.8	17.2	59.8	7.0	8.5	15.6	3.1	5.6	59.9	13.8	**80.8**	21.4	0.0	47.3	23.3	18.5	0.0	0.0	23.6
	AdaptSeg [7]†	63.9	17.9	60.7	9.6	15.0	16.8	6.5	11.5	61.2	15.3	78.5	24.4	14.4	53.4	18.3	14.5	0.0	3.6	25.6
	Ours (Stage-I)†	75.0	31.5	65.0	11.3	19.5	22.0	8.6	14.7	61.3	17.9	79.3	**29.6**	3.0	64.1	20.7	16.9	0.0	0.3	28.5
	Ours (Stage-II)†	**78.5**	**36.6**	**65.7**	**12.9**	**23.9**	**25.4**	**9.8**	**16.3**	**62.6**	16.8	80.7	29.1	0.0	**67.5**	**30.1**	23.2	0.0	1.7	**30.6**
Snowy	Source Only [10]	50.8	4.7	45.1	5.9	**24.0**	8.5	10.8	8.7	35.9	9.4	60.5	17.3	0.0	47.7	9.7	3.2	0.0	0.7	18.0
	AdaptSegNet [7], [10]	59.9	13.3	52.7	3.4	15.9	14.2	12.2	7.2	51.0	**10.8**	72.3	21.9	0.0	55.0	11.3	1.7	0.0	0.0	21.2
	CBST [10], [35]	59.6	11.8	57.2	2.5	19.3	13.3	7.0	9.6	41.9	7.3	70.5	18.5	0.0	61.7	8.7	1.8	0.0	0.2	20.6
	IBN-Net [10], [46]	61.3	13.5	57.6	3.3	14.8	17.7	10.9	6.8	39.0	6.9	71.6	22.6	0.0	56.1	13.8	20.4	0.0	0.0	21.9
	OCDA [10]	68.0	10.9	61.0	2.3	23.4	15.8	12.3	6.9	48.1	9.9	74.3	19.5	0.0	58.7	10.0	13.8	0.0	0.1	22.9
	MOCDA [16]†	71.8	16.9	61.1	6.5	21.4	16.3	17.0	7.5	52.9	8.7	79.7	29.2	0.5	62.7	18.9	**29.4**	0.0	**22.6**	27.5
	Source Only†	68.1	11.7	65.5	7.9	16.0	16.3	10.0	5.1	**55.0**	5.9	**81.6**	27.4	0.0	63.5	18.8	10.6	0.0	0.0	24.4
	AdaptSeg [7]†	65.3	12.6	68.6	15.6	19.8	17.6	17.7	11.6	51.0	6.8	79.3	35.3	**6.5**	63.5	15.7	21.2	0.0	9.4	27.2
	Ours (Stage-I)†	81.8	20.0	70.8	19.6	20.8	18.9	21.4	15.4	52.1	8.5	78.6	36.0	0.6	74.4	25.9	20.2	0.0	14.7	30.5
	Ours (Stage-II)†	**83.4**	**22.7**	**71.6**	**21.3**	21.9	**21.9**	**23.1**	**17.6**	54.2	9.2	80.8	**36.8**	0.0	**74.7**	**29.8**	28.9	0.0	15.9	**31.9**
Cloudy	Source Only [10]	47.0	8.8	33.6	4.5	20.6	11.4	13.5	8.8	55.4	25.2	78.9	20.3	0.0	53.3	10.7	4.6	0.0	0.0	20.9
	AdaptSegNet [7], [10]	51.8	15.7	46.0	5.4	25.8	18.0	12.0	6.4	64.4	26.4	82.9	24.9	0.0	58.4	10.5	4.4	0.0	0.0	23.8
	CBST [10], [35]	56.8	21.5	45.9	5.7	19.5	17.2	10.3	8.6	62.2	24.3	89.4	20.0	0.0	58.0	14.6	0.1	0.0	0.1	23.9
	IBN-Net [10], [46]	60.8	18.1	50.5	8.2	25.6	20.4	12.0	11.3	59.3	24.7	84.8	24.1	12.1	59.3	13.7	9.0	0.0	1.2	26.1
	OCDA [10]	69.3	20.1	55.3	7.3	24.2	18.3	12.0	7.9	64.2	27.4	88.2	24.7	0.0	62.8	13.6	18.2	0.0	0.0	27.0
	MOCDA [16]†	79.6	21.7	61.4	11.0	27.6	19.4	13.4	8.3	69.0	26.4	89.1	25.0	3.2	69.5	22.7	**21.5**	0.0	3.5	30.1
	Source Only†	70.1	16.0	64.1	8.5	26.9	17.6	9.3	7.6	69.5	23.5	87.0	25.7	0.0	66.1	26.6	8.9	0.0	0.0	27.8
	AdaptSeg [7]†	69.1	21.0	67.2	12.9	35.2	20.0	14.8	17.1	72.7	24.2	88.7	32.9	23.1	58.6	26.5	14.3	0.0	5.5	31.8
	Ours (Stage-I)†	85.2	30.9	69.1	20.3	34.6	21.4	15.9	20.4	72.8	30.4	88.9	**38.8**	32.4	77.3	33.6	8.4	0.0	11.4	36.4
	Ours (Stage-II)†	**86.1**	**35.7**	**69.9**	**21.3**	**36.9**	**24.5**	**16.9**	**23.0**	**73.7**	**31.0**	**89.9**	37.0	**33.1**	**78.0**	**36.5**	10.2	0.0	**11.6**	**37.6**
Overcast	Source Only [10]	46.6	9.5	38.5	2.7	19.8	12.9	9.2	17.5	52.7	19.9	76.8	20.9	1.4	53.8	10.8	8.4	0.0	1.8	21.2
	AdaptSegNet [7], [10]	59.5	24.0	49.4	6.3	23.3	19.8	8.0	14.4	61.5	22.9	74.8	29.9	0.3	59.8	12.8	9.7	0.0	0.0	25.1
	CBST [10], [35]	58.9	26.8	51.6	6.5	17.8	17.9	5.9	17.9	60.9	21.7	87.9	22.9	0.0	59.9	11.0	2.1	0.0	0.2	24.7
	IBN-Net [10], [46]	62.9	25.3	55.5	6.5	21.2	22.3	7.2	15.3	53.3	16.5	81.6	31.1	2.4	59.1	10.3	14.2	0.0	0.0	25.5
	OCDA [10]	73.5	26.5	62.5	8.6	24.2	20.2	8.5	15.2	61.2	23.0	86.3	27.3	0.0	64.4	14.3	13.3	0.0	0.0	27.9
	MOCDA [16]†	80.1	28.6	66.0	13.0	26.6	20.9	8.9	15.5	67.0	25.1	87.7	33.2	9.5	69.2	23.0	18.3	**2.2**	2.0	31.4
	Source Only†	72.9	23.3	68.8	10.1	19.7	18.8	6.2	11.3	69.0	23.1	87.5	36.1	10.5	67.8	26.3	9.4	0.0	0.0	29.5
	AdaptSeg [7]†	69.9	26.4	71.0	14.9	25.6	21.1	11.5	22.1	70.0	25.5	87.9	39.6	20.8	61.7	25.2	13.9	0.0	2.0	32.1
	Ours (Stage-I)†	85.1	38.3	73.5	25.3	29.0	24.5	12.4	26.2	70.9	**32.1**	88.3	**46.1**	**22.5**	**76.0**	31.0	**21.7**	0.7	7.2	37.4
	Ours (Stage-II)†	**86.0**	**41.2**	**73.9**	**25.7**	**30.6**	**27.7**	**13.6**	**27.4**	**71.9**	31.8	**89.3**	44.3	17.5	75.9	**37.0**	21.6	0.0	**7.4**	**38.0**

TABLE XII
COMPARISON OF DIFFERENT SELF-TRAINING METHODS.

Methods	Compound(C)			Open(O)	Avg	
GTA5 →	Rainy	Snowy	Cloudy	Overcast	C	C+O
Stage-I	28.5	30.5	36.4	37.4	32.8	33.2
MPT [60]	29.4	30.3	36.8	37.3	33.3	33.5
ProDA [8]	30.3	31.1	36.7	37.1	33.6	33.8
SFDA [11]	30.5	30.8	37.0	37.8	33.7	34.0
MPT [60]+CPSS (ours)	**30.6**	**31.9**	**37.6**	**38.0**	**34.4**	**34.5**

samples of the "train", "motorcycle" and ""bicycle" classes, which are the long tail classes rarely appearing in the C-Driving dataset.

F. Discussion

The improvement of our method in the second stage is not as significant as that in the first stage. To verify the effectiveness of our target adaptation stage, we compare our adaptation stage with advanced UDA [8], [60] and source-free DA [11] methods. Specifically, MPT [60] is our base self-training method, which generates pseudo-labels based on the pixel probability value and a predefined threshold. ProDA [8] is a state-of-the-art self-training technique in UDA, which utilizes the class prototype to optimize the pseudo-label maps. SFDA [11] is designed for single target adaptation under the source-free constraint, which transfers knowledge from the source distribution and trains the target model with patch-level self-supervision. As shown in Tab. XII, both MPT and ProDA fail to achieve consistent improvements on all domains (especially on the unseen domain). In addition, SFDA yields more improvement than the UDA self-training techniques but the improvement is still limited. Instead, with CPSS, the results on all domains are increased and our model achieves better performance on both compound and open domains. There are three main reasons accounting for this phenomenon: first, compared with single target (*e.g.*, CityScapes [64]) that previous domain adaptation methods [8], [11], [60] focus on, the compound and open target domains in OCDA are much more complex with more various scenes and situations; second, due to the complexity of target domain and the source-free constraint, the source

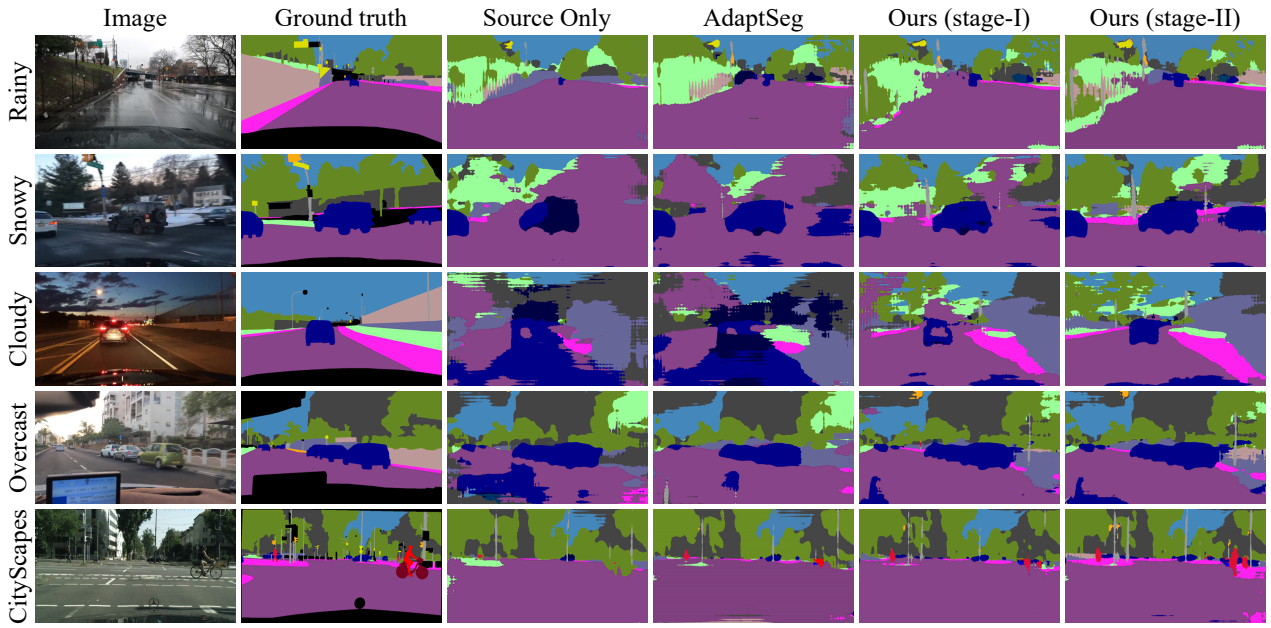


Fig. 5. Comparison of segmentation results on the compound (rainy, snowy, and cloudy) and open (overcast and CityScapes) domains.

Fig. 6. Examples of stylized images of CPSS. We directly apply CPSS on the image-level for image pairs on GTA5 (a and b) and C-Driving (c and d). The number of patches is set to 4.

model trained in the domain generalization manner is not good enough to provide accurate pseudo-labels; finally, the source-free constraint is very challenging for the existing self-training methods [8], [60] since the model can readily overfit to the noisy pseudo-labels without the guidance of labeled source data. In such a context, the existing self-training methods commonly cannot obtain high-quality pseudo-labels for the compound domain and do not consider the performance on the open domains, limiting their performance on SF-OCDA. Compared with them, our CPSS can alleviate the influence of noisy pseudo-labels and improve the generalization ability of the model to achieve better performance on both compound and open domains.

G. Visualization

Qualitative Comparison of Segmentation Results. We compare the segmentation results for different models on the compound domain (rainy, snowy, cloudy) and open domains (overcast and CityScapes) in Fig. 5. Compared with the source only model and AdaptSeg [7], our models (Stage-I and Stage-II) clearly produce more accurate semantic results, especially the boundaries between different objects. Comparing between our models, our Stage-II model can generate finer results on elements that have large intra-class variations between the virtual and real, *e.g.*, person, car and fence.

Image-Level Visualization of CPSS. To better understand the effect of our CPSS in style augmentation, we visualize four groups of style exchanging in Fig. 6 by applying CPSS in the image-level (*i.e.*, block-0). For each group, we feed two original images (left column) into CPSS and generate corresponding stylized images (right column) by swapping patch styles among the 8 (2×4) patches. We obverse that the styles of patches are successfully changed and various patches are generated. We can easily infer that CPSS can also change styles in the feature-level.

V. Conclusion

In this work, we introduce a new setting for semantic segmentation, called source-free open compound domain adaptation (SF-OCDA), which has a great potential in real-world applications. To address this challenging problem, we propose an effective framework to train robust source and target models under the source-free constraint. Moreover, the Cross-Patch Style Swap (CPSS) module is proposed to diversify the feature-level samples with various styles, which can consistently promote the results for both source and target training stages. Extensive experiments demonstrate the effectiveness of the proposed CPSS. Our method achieves state-of-the-art results on OCDA and domain generalization benchmarks.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015. 1, 2

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE TPAMI*, 2017, doi:10.1109/TPAMI.2017.2699184. 1, 2, 6

[3] W. Shi, J. Xu, D. Zhu, G. Zhang, X. Wang, J. Li, and X. Zhang, "Rgb-d semantic segmentation and label-oriented voxelgrid fusion for accurate 3d semantic mapping," *IEEE TCSVT*, 2021, doi:10.1109/TCSVT.2021.3056726. 1, 2

[4] X. Weng, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, "Stage-aware feature alignment network for real-time semantic segmentation of street scenes," *IEEE TCSVT*, 2021, doi:10.1109/TCSVT.2021.3121680. 1, 2

[5] J. Ji, R. Shi, S. Li, P. Chen, and Q. Miao, "Encoder-decoder with cascaded crfs for semantic segmentation," *IEEE TCSVT*, 2020, doi:10.1109/TCSVT.2020.3015866. 1, 2

[6] X. Sun, C. Chen, X. Wang, J. Dong, H. Zhou, and S. Chen, "Gaussian dynamic convolution for efficient single-image segmentation," *IEEE TCSVT*, 2021, doi:10.1109/TCSVT.2021.3096814. 1, 2

[7] 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. 1, 2, 3, 6, 7, 9, 10, 11

[8] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *CVPR*, 2021. 1, 2, 3, 10, 11

[9] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *IJCV*, 2021, doi:10.1007/s11263-020-01395-y. 1, 2, 3

[10] Z. Liu, Z. Miao, X. Pan, X. Zhan, D. Lin, S. X. Yu, and B. Gong, "Open compound domain adaptation," in *CVPR*, 2020. 1, 3, 4, 6, 10

[11] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *CVPR*, 2021. 1, 3, 9, 10

[12] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *ICCV*, 2019. 1, 3, 7

[13] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *CVPR*, 2019. 1, 3, 7

[14] Y. Tian and S. Zhu, "Partial domain adaptation on semantic segmentation," *IEEE TCSVT*, 2021, doi:10.1109/TCSVT.2021.3116210. 1

[15] K. Park, S. Woo, I. Shin, and I.-S. Kweon, "Discover, hallucinate, and adapt: Open compound domain adaptation for semantic segmentation," in *NeurIPS*, 2020. 1, 3, 6, 7, 8

[16] R. Gong, Y. Chen, D. P. Paudel, Y. Li, A. Chhatkuli, W. Li, D. Dai, and L. Van Gool, "Cluster, split, fuse, and update: Meta-learning for open compound domain adaptive semantic segmentation," in *CVPR*, 2021. 1, 3, 6, 8, 10

[17] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016. 1, 6

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 1, 4, 6

[19] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *ICML*, 2020. 1, 3

[20] J. N. Kundu, N. Venkat, R. V. Babu *et al.*, "Universal source-free domain adaptation," in *CVPR*, 2020. 1, 3

[21] P. T. S and F. Fleuret, "Uncertainty reduction for model adaptation in semantic segmentation," in *CVPR*, 2021. 1, 3, 9

[22] Z. Li, Y. Sun, L. Zhang, and J. Tang, "Ctnet: Context-based tandem network for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, doi:10.1109/TPAMI.2021.3132068. 2

[23] Y. Sun and Z. Li, "Ssa: Semantic structure aware inference for weakly pixel-wise dense predictions without cost," *arXiv preprint arXiv:2111.03392*, 2021. 2

[24] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, "Weakly-supervised semantic segmentation via sub-category exploration," in *CVPR*, 2020. 2

[25] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *CVPR*, 2018. 2

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016. 2

[27] F. Cermelli, M. Mancini, S. R. Bulo, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *CVPR*, 2020. 2

[28] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "Plop: Learning without forgetting for continual semantic segmentation," in *CVPR*, 2021. 2

[29] A. Maracani, U. Michieli, M. Toldo, and P. Zanuttigh, "Recall: Replay-based continual learning in semantic segmentation," in *ICCV*, 2021. 2

[30] T. Kalb, M. Roschani, M. Ruf, and J. Beyerer, "Continual learning for class- and domain-incremental semantic segmentation," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, doi:10.1109/IV48863.2021.9575493. 2

[31] D. Shenaj, F. Barbato, U. Michieli, and P. Zanuttigh, "Continual coarse-to-fine domain adaptation in semantic segmentation," *Image and Vision Computing*, 2022, doi:10.1016/j.imavis.2022.104426. 2

[32] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *CVPR*, 2019. 3

[33] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *ICCV*, 2019. 3

[34] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann, "Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation," *NeurIPS*, 2020. 3

[35] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *ECCV*, 2018. 3, 6, 7, 10

[36] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," in *IJCAI*, 2020. 3, 4

[37] K. Mei, C. Zhu, J. Zou, and S. Zhang, "Instance adaptive self-training for unsupervised domain adaptation," in *ECCV*, 2020. 3

[38] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *CVPR*, 2020. 3

[39] H. Ma, X. Lin, Z. Wu, and Y. Yu, "Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization," in *CVPR*, 2021. 3

[40] Z. Zheng and Y. Yang, "Adaptive boosting for domain adaptation: Towards robust predictions in scene segmentation," *arXiv preprint arXiv:2103.15685*, 2021. 3

[41] J. Huang, D. Guan, A. Xiao, and S. Lu, "Fsdr: Frequency space domain randomization for domain generalization," in *CVPR*, 2021. 3

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009. 3, 7

[43] W. Chen, Z. Yu, Z. Wang, and A. Anandkumar, "Automated synthetic-to-real generalization," in *ICML*, 2020. 3, 7

[44] W. Chen, Z. Yu, S. De Mello, S. Liu, J. M. Alvarez, Z. Wang, and A. Anandkumar, "Contrastive syn-to-real generalization," in *ICLR*, 2021. 3

[45] S. Choi, S. Jung, H. Yun, J. Kim, S. Kim, and J. Choo, "Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening," in *CVPR*, 2021. 3

[46] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *ECCV*, 2018. 3, 6, 7, 10

[47] Z. Tang, Y. Gao, Y. Zhu, Z. Zhang, M. Li, and D. Metaxas, "Selfnorm and crossnorm for out-of-distribution robustness," in *ICCV*, 2021. 3, 7, 8

[48] I. Kuzborskij and F. Orabona, "Stability and hypothesis transfer learning," in *ICML*, 2013. 3

[49] B. Chidlovskii, S. Clinchant, and G. Csurka, "Domain adaptation in the absence of source domain data," in *ACM KDD*, 2016. 3

[50] J. Liang, R. He, Z. Sun, and T. Tan, "Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation," in *CVPR*, 2019. 3

[51] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE TPAMI*, 2021, doi:10.1109/TPAMI.2021.3103390. 3

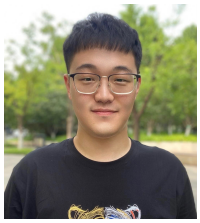
[52] J. Tian, J. Zhang, W. Li, and D. Xu, "Vdm-da: Virtual domain modeling for source data-free domain adaptation," *IEEE TCSVT*, 2021, doi:10.1109/TCSVT.2021.3111034. 3

[53] Y. Hou and L. Zheng, "Visualizing adapted knowledge in domain transfer," in *CVPR*, 2021. 3

[54] S. Stan and M. Rostami, "Unsupervised model adaptation for continual semantic segmentation," in *AAAI*, 2021. 3

[55] J. N. Kundu, A. Kulkarni, A. Singh, V. Jampani, and R. V. Babu, "Generalize then adapt: Source-free domain adaptive semantic segmentation," in *ICCV*, 2021. 3, 4

[56] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, 2008. 4
[57] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016. 4
[58] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *ICLR*, 2017. 4
[59] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017. 5
[60] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *CVPR*, 2019. 5, 10, 11
[61] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020. 6
[62] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *NeurIPS*, 2021. 6
[63] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*, 2020. 6
[64] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016. 6, 10
[65] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *ICCV*, 2019. 6
[66] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016. 6
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 6
[68] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *ICCV*, 2019. 7
[69] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *ICLR*, 2021. 7, 8



Yuyang Zhao received his B.E. degree in Electronic Information Engineering from Tianjin University, China, in 2020. He is currently pursuing the Ph.D. degree in Computer Science from National University of Singapore, Singapore. His research interests include semantic segmentation, domain adaptation and domain generalization.

Zhun Zhong received his Ph.D. degree in the Department of Artificial Intelligence of Xiamen University, China, in 2019. He was also a joint Ph.D. student at University of Technology Sydney, Australia. He is now a postdoc at University of Trento, Italy. His research interests include person re-identification, novel class discovery, data augmentation and domain adaptation.

Zhiming Luo received the B.S. degree from the Cognitive Science Department, Xiamen University, Xiamen, China, in 2011; the Ph.D. degree in computer science with Xiamen University and University of Sherbrooke, Sherbrooke, QC, Canada, in 2017. His research interests include surveillance video analytic, computer vision, and machine learning.

Gim Hee Lee is currently an Associate Professor at the Department of Computer Science at the National University of Singapore (NUS), where he heads the Computer Vision and Robotic Perception (CVRP) Laboratory. He is also affiliated with the NUS Graduate School for Integrative Sciences and Engineering (NGS), and the NUS Institute of Data Science (IDS). He was a researcher at Mitsubishi Electric Research Laboratories (MERL), USA. Prior to MERL, he did his PhD in Computer Science at ETH Zurich. He received his B.Eng and M.Eng degrees from the Department of Mechanical Engineering at NUS. He has served as Area Chair for major computer vision conferences such as CVPR, ICCV, ECCV, BMVC, 3DV, WACV, and will be part of the organizing committee as one of the Program Chairs for 3DV 2022 and the Exhibition/Demo Chair for CVPR 2023.

Nicu Sebe is Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a general chair of ACM Multimedia 2022 and a program chair of ECCV 2024. He is a fellow of IAPR.