Both Style and Fog Matter: Cumulative Domain Adaptation for Semantic Foggy Scene Understanding

Xianzheng Ma[1,2], Zhixiang Wang[3,4], Yacheng Zhan[1], Yinqiang Zheng[3], Zheng Wang[1,2*]
Dengxin Dai[5], Chia-Wen Lin[6]

[1]National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science, Wuhan University [2]Hubei Key Laboratory of Multimedia and Network Communication Engineering
[3]The University of Tokyo [4]RIISE [5]MPI for Informatics [6]National Tsing Hua University

Abstract

Although considerable progress has been made in semantic scene understanding under clear weather, it is still a tough problem under adverse weather conditions, such as dense fog, due to the uncertainty caused by imperfect observations. Besides, difficulties in collecting and labeling foggy images hinder the progress of this field. Considering the success in semantic scene understanding under clear weather, we think it is reasonable to transfer knowledge learned from clear images to the foggy domain. As such, the problem becomes to bridge the domain gap between clear images and foggy images. Unlike previous methods that mainly focus on closing the domain gap caused by fog — defogging the foggy images or fogging the clear images, we propose to alleviate the domain gap by considering fog influence and style variation simultaneously. The motivation is based on our finding that the style-related gap and the fog-related gap can be divided and closed respectively, by adding an intermediate domain. Thus, we propose a new pipeline to cumulatively adapt style, fog and the dual-factor (style and fog). Specifically, we devise a unified framework to disentangle the style factor and the fog factor separately, and then the dual-factor from images in different domains. Furthermore, we collaborate the disentanglement of three factors with a novel cumulative loss to thoroughly disentangle these three factors. Our method achieves the state-of-the-art performance on three benchmarks and shows generalization ability in rainy and snowy scenes.

1. Introduction

Semantic foggy scene understanding (SFSU) is important for autonomous driving [9, 14, 18, 27, 28, 40]. Although great progress has been made on semantic understanding of clear scenes, SFSU tends to have unsatisfactory performance due to visibility degradation caused by fog [22, 31]. Besides, unlike the abundant data and annotation under clear scenes, the lack of data and annotation under dense fog weather further complicates this problem. Therefore, handling the challenging SFSU problem often requires transferring the segmentation knowledge learned from labeled *clear* images to the unlabeled *foggy* images.

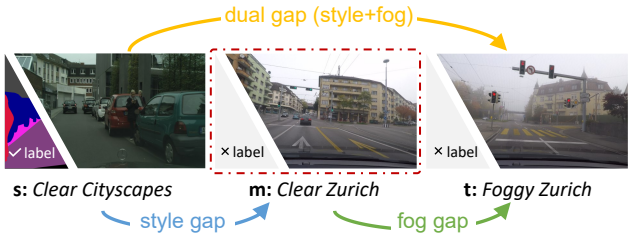

Figure 1. **The problem and our main idea.** Our goal is to transfer the knowledge from a labeled domain s to an unlabeled domain t. However, direct knowledge transfer is challenging due to the mixed dual-factor gap (orange arrow). By adding an intermediate domain m as a bridge, we can decompose the mixed dual-factor gap into two single-factor gaps: the style gap and the fog gap. Since images in both domains s and m are captured in clear scenes, we assume there is only the style gap between domains s and m (blue arrow). Likewise, images in both domains m and t are collected in the same city (Zurich), we assume there exists only the fog gap between them (green arrow).

Intuitively, we may tackle this problem by closing the domain gap between clear images and foggy images with state-of-the-art domain adaptation methods. However, these methods mainly *align* domains in an adversarial [4, 6, 17, 19, 33–37, 41] or a self-training [2, 7, 11, 13, 21, 32, 39] manner, regardless of how the domain gap is caused. Besides, as has been validated by [27], they fail to address the SFSU problem well due to the *large* domain gap.

Consequently, the attention of SFSU has been focused on the fog factor, which is regarded as the dominating cause of the domain gap in the SFSU problem. One solution is to

*Corresponding Author

close this gap by defogging real foggy images, using an existing defogging method [3, 10, 15, 24–26, 38]. Whereas, the defogging method will also introduce artifacts. They act as noise to hinder the domain adaptation to some extent [23]. Another solution is to add synthetic fog to clear images and learn with these synthetic foggy images and annotations of clear images in a supervised manner [9, 12, 14, 27, 28]. Nevertheless, these rendered synthetic foggy images, not as real as real foggy ones, could also widen the domain gap between clear and foggy images and yield unsatisfactory performance. Moreover, we argue that these methods overconcerned the fog factor while ignoring other factors, which may affect the domain gap in the SFSU problem.

Thinking out of the box, we propose to explicitly investigate the domain gap in SFSU **1)** to avoid directly treating the total domain gap; **2)** without using synthetic foggy data or knowledge of defogging. We assume that the domain gap is caused by the mixed fog influence and style variation, and both of them are important to SFSU. That is, we assume there exist the style-related gap and the fog-related gap in the domain gap of SFSU, and we can decompose the mixed dual-factor gap into these two single-factor gaps by adding an intermediate domain. Next, we elaborate on why we can disentangle the style-related gap and what relationship lies between the style-related gap and the fog-related gap in the SFSU problem, using the following empirical finding.

1.1. Motivation

We first investigate the influence of the style and fog factors across different domains, *i.e.*, we want to know how the style and fog factors affect the performance of a segmentation model. To this end, as shown in Figure 2, we utilize Mean Variance Value (MVV) to represent how a segmentation model functions in each domain and how the gap between two different domains is closed. As has been validated by [42], the variance, which is calculated from different-level features in a segmentation model, has a strong ability in measuring the uncertainty of a segmentation model when predicting pixel labels. We obtain one Variance Value to represent the uncertainty when the model segments one image. Thus, we calculate the MVV of all images in a specific domain dataset to show the overall performance in this domain.

Specifically, in Figure 2, we trained a segmentation model Model (s) with s-domain data and calculate MVV in domains s, m and t, yielding V_s^s, V_s^m and V_s^t, respectively. We use the length of the bar to indicate the performance in each domain. Ideally, since we only have the model learned from domain s, its performance should be good when segmenting images in domain s (*i.e.*, MVV should be low), but tends to degrade when segmenting images in domain m and t (*i.e.*, MVV should be relatively high). Our experiments results are as expected and the yellow bar becomes cumulatively longer when dealing with the images in domain s, m, and t. Besides, we can use the difference of two MVVs as the performance gap between two domains (*i.e.*, domain gap). For example, $V_s^m - V_s^s$ can represent the gap between domain s and m, which we assume as "style gap". Likewise, we obtain the "dual gap" and "fog gap".

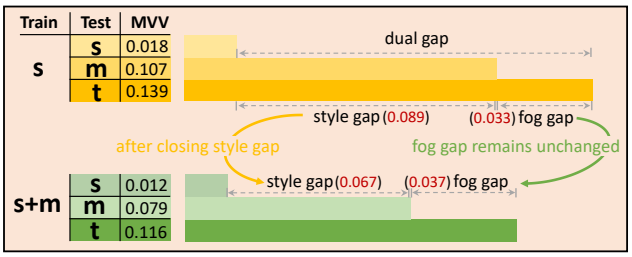

Figure 2. **Empirical finding of the motivation.** The mean variance value (MVV) measures the overall performance of the segmentation model in a specific domain *i.e.*, domain performance. At first, we train a segmentation model with s domain data *i.e.*, this model has learned s domain knowledge. Then, we test it on s, m and t domain data, and the performances in three domains are shown as different yellow bars. Besides, the difference between two different bars can represent the performance gap *i.e.*, domain gap (gray dotted arrow), such as the style gap, fog gap and dual gap. Next, we adapt the segmentation model with m-domain data *i.e.*, this model can learn domain knowledge (related to the style factor) between domains s and m, which means the style gap can be closed by this adaptation. After this adaptation, the style gap has been closed (from 0.089 to 0.067) while the fog gap remains unchanged (a negligible change of only 0.004). That is, by adding an intermediate domain m, we disentangle the style gap from the dual gap without damaging the fog gap. Thus, we assume that the style gap and fog gap can be divided and closed respectively and the dual gap is an accumulation of the two gaps.

Then, we adapt Model (s) with m-domain data to obtain Model ($s + m$) and calculate the MVV in three domains. Compared with Model (s), Model ($s+m$) can learn domain knowledge (related to the style factor) between domains s and m and thus close the style gap (from 0.089 to 0.067). However, the fog gap (0.037) remains large and approximately equals to the fog gap (0.033) before adapting Model (s) with m-domain data. That is, after closing the style gap, the fog gap still remains unchanged, which means the two gaps can be divided and closed respectively. Meanwhile, the dual gap is always an accumulation of the style gap and the fog gap before and after this adaptation.

Based on this finding, we propose a cumulative domain adaptation framework to address semantic foggy scene understanding, considering both style factor and fog factor in this task. As shown in Figure 1, by adding an intermediate domain m as a bridge, we can decompose the mixed dual-factor gap into two single-factor gaps: the style gap and the fog gap. Specifically, we disentangle the style and fog factor separately, and then the dual-factor (style and fog) jointly,

which ensures an effective segmentation knowledge transfer from the source domain to the target domain. Besides, we assume that the dual-factor gap is an accumulation of the style gap and the fog gap. Thus, we further propose a novel cumulative loss to represent this relationship and collaborate the disentanglement of three factors with the cumulative loss in a cyclical manner, enabling our network to transfer segmentation knowledge continuously and further improving the performance.

We summarize our contributions as follows. **1)** We devise a novel framework, which disentangles the dual-factors gap in SFSU into two single-factor and smaller gaps (style gap and fog gap). Specifically, we propose a novel Cumulative Domain Adaptation (CuDA-Net) method, first disentangling the style factor and fog factor separately, and then the dual-factor jointly. **2)** We find the cumulative relationship of style, fog, and dual factors and thus propose a novel cumulative loss to further disentangle the three factors in a cyclical manner. **3)** Our method outperforms state-of-the-arts on three widely used datasets in SFSU and shows generalization ability on other adverse scenes, such as rainy and snowy scenes.

2. Method

Suppose that we have N_s labeled images $\{(x_s^i, y_s^i)\}_{i=1}^{N_s}$ from the source domain s, where y_s^i is the label, and N_t unlabeled images $\{x_t^i\}_{i=1}^{N_t}$ from the target domain t. Our goal is to transfer the segmentation knowledge from the source domain s to the target domain t by our proposed CuDA-Net. Motivated by the success of [4], we use a similar framework as our basic unit to disentangle domain-invariant features from the domain-specific counterparts. However, since images in domain s and t are taken under different cities and weathers, they encounter large domain gaps caused by mixed style and fog factors, which challenges this method. Therefore, we propose to decompose the mixed factors into separate ones by introducing an intermediate domain m with N_m unlabeled images $\{x_m^i\}_{i=1}^{N_m}$, which share similar fog influence (no fog) with the source domain and similar style variation with the target domain (same city). Figure 3 depicts the framework of our proposed method. It includes three sub-networks: $F_{s\to m}$, $F_{m\to t}$ and $F_{s\to t}$, which share the same prototypes to disentangle the domain-invariant features from the domain-specific counterparts (Figure 3a). They are fed with different input pairs (x_s, x_m), (x_m, x_t) and (x_s, x_t) to close the style gap, the fog gap and the dual gap respectively. We train them one by one (Figure 3b) and share the domain-invariant knowledge forward. After training these three sub-networks (initialization), we conduct a cyclical training (Figure 3d) using the cumulative relation (Figure 3c) as auxiliary loss, to help better disentangle the domain-invariant (content) features, which are used to produce segmentation heatmaps.

2.1. Feature Disentanglement Networks

Feature Disentanglement Networks (FDN) is the basic unit of our method, as shown in Figure 3a. Given images x_1 and x_2 from two different domains, with the "shared content space" assumption [16], it can disentangle domain-invariant content features c_1 and c_2 of these images from the domain-specific counterparts z_1 and z_2. As has been validated by [4], the content features contributes most to the semantic segmentation task. Therefore, through feature disentanglement, we can transfer the segmentation knowledge from x_1 domain to x_2 domain.

Specifically, we first use a shared content encoder E_c (black line) to extract c_1 and c_2 and two private encoders to extract domain-specific feature z_1 and z_2 respectively (red and blue line). Then, we use an shared image decoder D to decode an image using the content features c_1, c_2, and domain-specific feature z_1, z_2. Depending on which c and z we use, we can perform within-domain reconstruction and cross-domain translation to supervise the disentanglement learning. Besides, we use a segmentation head S to produce segmentation heatmaps h from the content feature c, where label y_1 is used as the supervision signal.

We build our FDN with a similar framework as DISE [4] because both of us adopt the "shared content space" assumption [16]. However, we only design four necessary losses to train our FDN, aiming to enable the FDN to close three different gaps (style gap, fog gap and dual gap). While, DISE [4] utilizes seven losses to close one gap between synthetic clear data and real clear data, which is time-consuming to train and hard to converge.

Within-domain reconstruction. We expect images decoded using content feature c and private feature z extracted from the same image can perfectly reconstruct the original ones. Thus, we define the reconstruction loss as:

$$L_{rec} = L_{pixel}(x_1, \hat{x}_1) + L_{pixel}(x_2, \hat{x}_2)\,, \tag{1}$$

where the pixel-wise loss $L_{pixel}(\cdot,\cdot)$ is implemented by the perceptual loss [30] with shallow layer features highlighted.

Cross-domain translation. We recombine the content feature c from one domain image and private feature z from another domain image to generate the translated image. For example, in our sub-network $F_{m\to t}$ in Figure 3b, by recombining the content feature c_t^2 from x_t and the private feature z_m^2 from x_m, we can generate an image, which can be regarded as the defogged version of x_t. For the translated images $x_{1\to 2}$ and $x_{2\to 1}$ whose private features have been changed, we impose content consistency losses L_{con}, which is implemented by the perceptual loss [30] with the deep layer features highlighted, to constrain the content aspect of translated images and original images:

$$L_{trans} = L_{con}(x_1, x_{1\to 2}) + L_{con}(x_2, x_{2\to 1})\,. \tag{2}$$

Dense pixel prediction. Thanks to the domain invariance discovered by disentanglement learning, we can transfer the

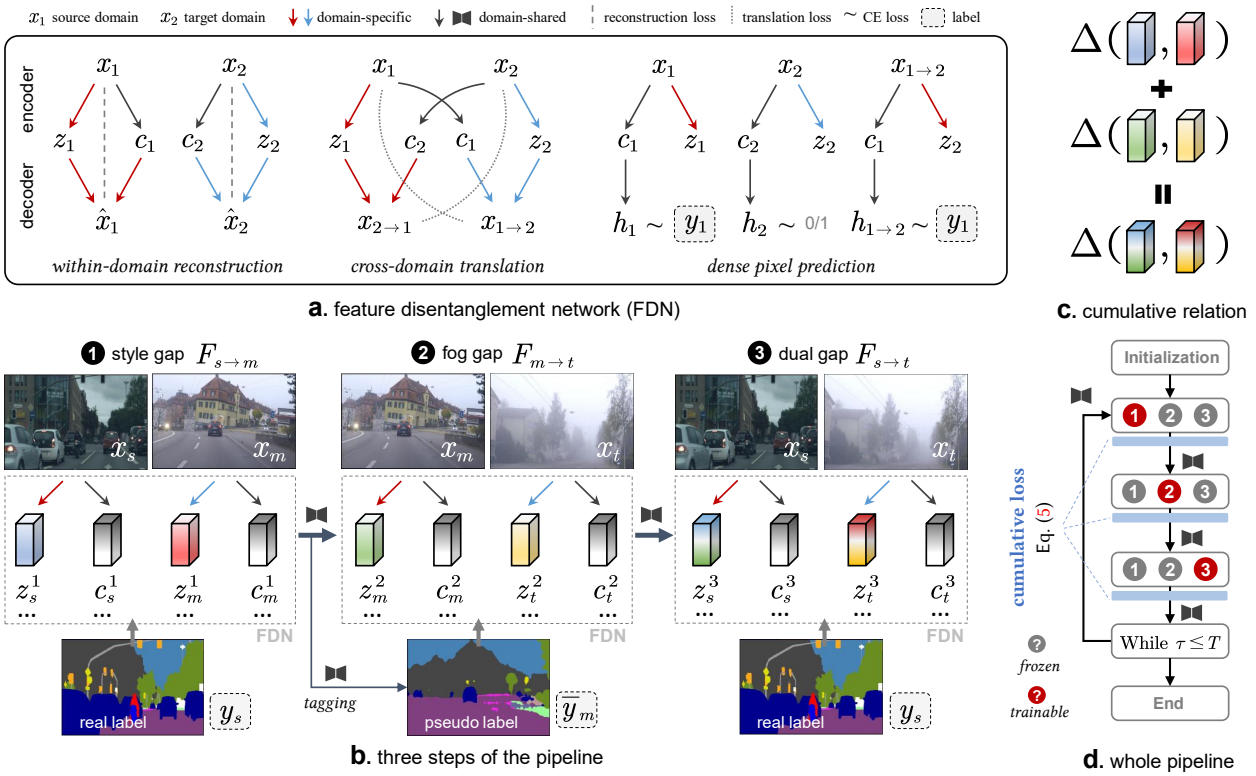

Figure 3. **The proposed method**. **a**, The feature disentanglement network (FDN) disentangles domain-invariant content features from the domain-specific counterparts for images from two different domains. **b**, By introducing the intermediate domain m, we can obtain three different input domain combinations, (x_s, x_m), (x_m, x_t) and (x_s, x_t), for three FDNs, $F_{s\to m}$, $F_{m\to t}$ and $F_{s\to t}$, to tackle the style gap, the fog gap and the dual gap respectively. Three FDNs are trained one by one, where the domain-invariant knowledge is shared. As there are no labels for both domain m and t, we use $F_{s\to m}$ to tag domain m for training $F_{m\to t}$. **d**, The whole pipeline. We first initialize three FDNs by training each of them once, as in **b**. Then, we conduct cyclical training, using the cumulative relation (**c**) as an auxiliary loss, for better disentangling the domain-invariant (content) features, which are used to produce segmentation heatmaps.

semantic knowledge across domains. We apply the segmentation head S on c_1 and c_2 to obtain the probability outputs of each pixel $h_1, h_2 \in \mathbb{R}^{H\times W\times C}$, where H, W, C represents the height, the width and the number of class categories, respectively. To supervise the training of the shared content encoder E_c and the segmentation head S, we use the cross-entropy to calculate segmentation loss L^1_{seg} between h_1 and its corresponding label y_1. Besides, since the 1-domain-like image $x_{1\to 2}$ share the same content as x_1, the labels y_1 can be the pseudo labels for $x_{1\to 2}$. Hence, we calculate $L^{1\to 2}_{seg}$ between $h_{1\to 2}$ and label y_1, also using the cross-entropy.

Aside from supervised losses, an adversarial loss L_{segadv} at the output of the segmentation head S is introduced, in the hopes of making the content encoder E_c and the S generalize well on the domain 2. To this end, we introduce the domain discriminator Dis and fool Dis by maximizing the probability of target domain prediction h_2 being considered as the source domain prediction:

$$L_{segadv} = -\sum_{h,w} \log\left(Dis\left(h_2\right)^{(h,w,1)}\right), \tag{3}$$

where 1 means the discriminator Dis perceive h_2 as the source domain prediction.

Feature disentanglement loss. The disentanglement loss function in FDN is a weighted combination of each loss:

$$\begin{aligned} L_{1\to 2} =& \lambda_{rec} L_{rec} + \lambda_{trans} L_{trans} \\ &+ \lambda_{seg}(L^1_{seg} + L^{1\to 2}_{seg}) + \lambda_{segadv} L_{segadv} \end{aligned}, \tag{4}$$

where $L_{1\to 2}$ can be $L_{s\to m}$, $L_{m\to t}$ or $L_{s\to t}$ for the following disentanglement and the weights λ_{rec}, λ_{trans}, λ_{seg} and λ_{segadv} are empirically set as 0.5, 0.1, 1 and 1 to control the relative importance of reconstruction/translation quality, the prediction accuracy and domain generalization.

2.2. Style and Fog Decomposition

The aforementioned FDN is designed to transfer the segmentation knowledge by disentangling the domain-invariant features and domain-specific features. However, directly applying FDN to domain s and domain t cannot achieve ideal performance. The reason we suppose is that the mixed dual-factor gap between domain s and domain t is too large to close, which is also the weakness of other domain adaptation methods. Thus, we introduce the inter-

mediate domain m, decomposing the dual-factor gap into two single-factor gaps: style and fog. Since images in both domain s and m are captured in clear scenes, we assume there is only the style gap between them. Similarly, as images in both domain m and t are collected in the same city (Zurich), we assume there exists only the fog gap between them. Therefore, we use three sub-network $F_{s\to m}$, $F_{m\to t}$ and $F_{s\to t}$ to disentangle the style factor, fog factor and dual-factor one by one and gradually transfer the segmentation knowledge from domain s to t (Figure 3b).

Concretely, $F_{s\to m}$ first utilizes two specific private style encoders E^s_{sty} and E^m_{sty} to extract the latent style features z^1_s and z^1_m, respectively. The labels $\{y^i_s\}^{N_s}_{i=1}$ supervise the training process. After then, except for the two private style encoders, the remaining part of this trained $F_{s\to m}$, which we perceive as domain shared part and represents segmentation knowledge, will be passed to the next sub-network $F_{m\to t}$. In other words, an content encoder E_c, segmentation head S, image decoder D and domain discriminator Dis are used as initialization of sub-network $F_{m\to t}$. Note that the domain m has no labels, we used the trained $F_{s\to m}$ to generate pseudo labels for training the $F_{m\to t}$. After training $F_{m\to t}$, except for the two fog encoders E^m_{fog} and E^t_{fog}, the domain shared part of $F_{m\to t}$ are used as the initialization of sub-network $F_{s\to t}$. Likewise, $F_{s\to t}$ uses two dual-factor (style and fog) encoders E^s_{dual} and E^t_{dual} to extract the latent dual-factor features z^3_s and z^3_t, respectively. To put it simply, through the training of $F_{s\to m}$, $F_{m\to t}$ and $F_{s\to t}$, we pass down the segmentation knowledge from domain s to domain t in a more effective way and obtain three pairs of domain-specific feature encoders for further feature disentanglement in the cumulative domain adaptation.

2.3. Cumulative Domain Adaptation

Cumulative loss. As verified in our motivation, there exists a cumulative relationship among three kinds of domain factors (private features). As shown in Figure 3c, if we take $\Delta(z_m, z_s)$ as the style discrepancy between domain m and s, take $\Delta(z_t, z_m)$ as the fog discrepancy between domain t and m, and take $\Delta(z_t, z_s)$ as the dual discrepancy between domain t and s, it is reasonable to assume that the dual discrepancy is a cumulation of the style and fog discrepancies, namely, $\Delta(z_m, z_s) + \Delta(z_t, z_m) = \Delta(z_t, z_s)$. Thus, we design the cumulative relationship loss function as:

$$L_{cum} = \|\Delta(z_m, z_s) + \Delta(z_t, z_m) - \Delta(z_t, z_s)\|^2 . \quad (5)$$

Then, we take a step further and use this cumulative loss L_{cum} as an additional loss to conduct our proposed cumulative domain adaptation, by utilizing private encoders trained in the first three steps before.

Training pipeline. Figure 3d depicts the whole training process. The three trained sub-network $F_{s\to m}$, $F_{m\to t}$ and $F_{s\to t}$ are used as the initialization of the cumulative domain adaptation. Specifically, we use all modules in $F_{s\to t}$ (an content encoder, two dual-factor encoders, an image decoder and an segmentation head), two style encoders in $F_{s\to m}$ and two fog encoders in $F_{m\to t}$ to build up the whole network. Next, as shown in Figure 3**b**, we input (x_s, x_m, x_t) tuples for extracting style, fog and dual private features. Then, we freeze the two fog encoders and two dual-factor encoders and train other modules (especially two style encoders and content encoders) in the whole network, using the final loss below:

$$L_{final} = L_{s\to m} + \lambda_{cum} L_{cum} . \quad (6)$$

After this training, we assume the style encoders can capture domain-specific style features better due to combining feature disentanglement loss $L_{s\to m}$ with the cumulative loss L_{cum} and the content encoders can better extract shared content features, which is used to produce the segmentation heatmap. The following two steps in Figure 3d have the same function and the only difference is which private encoders we train and which encoders we freeze. Note that the shared content encoder is always trainable in the three steps, and we use the content encoder to update pseudo labels for training fog encoders. Besides, we train the whole network in a cyclical manner, hoping to improve the disentangling ability of three pairs of private encoders alternatively and continuously enhance the shared content encoder. Empirically, we set T as 3, which means we conduct the cyclical cumulative training three times. Finally, we use the trained content encoder and segmentation head S in $F_{s\to t}$ to produce the segmentation heatmaps for testing.

3. Experiments

3.1. Datasets

Cityscapes [8] is a real-world dataset composed of street view images captured in 50 different cities. Its data split includes 2,975 training images and 500 validation images.
Foggy Cityscapes DBF [28] has 550 synthetic foggy images in total, including 498 training images and 52 testing images. These images are selected from Cityscapes and synthesized with fog using depth information. We use 498 clear images from Cityscapes as the source domain dataset, named as **Clear Cityscapes**. Note that images in Clear Cityscapes are not captured in Zurich city.
Foggy Zurich* [27] contains 3,808 real-world foggy road scenes in the city of Zurich and its suburbs. According to fog density, it is split into two categories—light and medium, consisting of 1,552 images and 1,498 images, respectively. We use the medium category as the target domain dataset, named as **Foggy Zurich**. Besides, it has a test set–Foggy Zurich-test including 40 images with labels that are compatible with Cityscapes [8].

Table 1. **Performance comparison**. Experiments are conducted on Foggy Zurich (FZ) and Foggy Driving (FD), measured with mean IoU (mIoU %) over all classes. For results on ACDC, please refer to the ACDC-fog benchmark website.

Experiment	Method	Backbone	FZ	FD
Backbone	–	DeepLab-v2	25.9	35.7
	–	RefineNet	34.6	35.8
Defogging	MSCNN [24]	RefineNet	34.4	38.3
	DCP [15]	RefineNet	31.2	33.2
	Non-local [3]	RefineNet	27.6	32.8
	GFN [25]	DeepLab-v2	27.5	37.2
	DCPDN [38]	DeepLab-v2	28.7	37.9
Domain Adaptation	Multi-task [1]	–	26.1	31.6
	AdSegNet [34]	DeepLab-v2	26.1	37.6
	ADVENT [35]	DeepLab-v2	24.5	36.1
	DISE [4]	DeepLab-v2	40.7	45.2
	CCM [19]	DeepLab-v2	35.8	42.6
	SAC [2]	DeepLab-v2	37.0	43.4
	ProDA [39]	DeepLab-v2	37.8	41.2
	DMLC [13]	DeepLab-v2	33.5	32.6
	DACS [32]	DeepLab-v2	28.7	35.0
Defogging+DA	MSCNN [24]+DISE [4]	DeepLab-v2	38.6	37.1
Ours	CuDA-Net	DeepLab-v2	**48.2**	**52.7**
Synthesis†	SFSU [28]	RefineNet	35.7	35.9
	CMAda2 [27]	RefineNet	42.9	37.3
	CycleGAN [43]	RefineNet	40.5	47.7
	MUNIT [16]	RefineNet	39.1	47.8
	AnalogicalGAN [12]	RefineNet	42.3	47.5
	CMAda3+ [9]	RefineNet	46.8	49.8
Synthesis+DA	SFSU [28]+DISE [4]	DeepLab-v2	39.3	39.0
Ours	CuDA-Net+	DeepLab-v2	**49.1**	**53.5**

†Since the synthesis-based methods use additional synthetic data, for fair comparison, we also add these data to train our sub-network $F_{m\to t}$ before cumulative domain adaptation, named CuDA-Net+.

Foggy Driving [27]. It is a collection of 101 real-world foggy road-scenes images, in which 33 images are finely annotated and the rest 68 images are coarsely annotated. They are purely used for testing.

Clear Zurich. We manually select 248 images from the light category of Foggy Zurich* [27] and term this dataset as Clear Zurich. We use this **Clear Zurich** as an intermediate domain dataset because we perceive these images visually as clear scene images.

ACDC [29]. It contains four adverse-condition categories (fog, rain, snow and nighttime) with pixel-level annotations. Each of them contains 1,000 images and is split into train set, validation set and test set for roughly 4:1:5 proportion. The test set is withheld for testing online.

3.2. Performance Comparison

We compare our method against several kinds of methods, including **1)** *backbones*: RefineNet [20] and DeepLab-v2 [5]; **2)** *defogging-based*: MSCNN [24], DCP [15], Non-local [3], DCPDN [38] and GFN [25]; **3)** *DA-based*: Multi-task [1], AdSegNet [34], ADVENT [35], CCM [19], SAC [2], ProDA [39], DMLC [13], DACS [32] and our baseline DISE [4]; **4)** *synthesis-based*: SFSU [28], CMAda2 [27], CMAda3+ [9], CycleGAN [43], MUNIT [16], AnalogicalGAN [12]. **5)** *Defogging/Synthesis + DA-based*: MSCNN+DISE, SFSU+DISE. The mean Intersection-Over-Union (mIoU) results on Foggy Zurich and Foggy Driving are reported in Table 1.

Table 2. **Training data comparison with CMAda3+.** Both our CuDA-Net and CuDA-Net+ outperform CMAda3+, using less synthetic foggy data and less real foggy data. 'light', 'medium' and 'dense' in the table indicates the different fog density.

Training data used	Fog density	CMAda3+	CuDA-Net	CuDA-Net+
Clear Cityscapes		498	498	498
Foggy Cityscapes DBF (synthetic fog)	light	498	–	–
	medium	498	–	–
	dense	498	–	498
Foggy Zurich* (real fog)	light	1552	248	248
	medium	1498	1498	1498
Total Number		5042	2244	2742
mIoU (on FZ)		**46.8**	**48.2**	**49.1**

For *Defogging-based* methods, we first use these methods to defog the real foggy test images and then use the backbone segmentation model to produce the predictions. For *Domain Adaptation based* methods, we set the source domain data of as clear cityscapes, s domain of our method. As for the target domain data, we combine the Clear Zurich and Foggy Zurich, which are used as the m domain and t domain in our method. By using the same amount of training data, we ensure a fair comparison with *DA-based* methods. For *Defogging+Domain Adaptation* methods, we first use the MSCNN [24] to defog the target domain data (including training data and test data) and then use DISE [4] to bridge the domain gap.

For *Synthesis-based* methods, the paradigm is to finetune the segmentation model pretrained on the real clear weather images (Cityscapes) with synthetic foggy images, e.g., Foggy Cityscapes DBF, and labels corresponding to its clear weather images. The difference in these *Synthesis-based* methods is that they use different methods [12,16,27,28,43] to generate the synthetic foggy images. Finally, the fine-tuned model is tested on real foggy images. For a fair comparison with CMAda3+ [9], we also add Foggy Cityscapes DBF as extra data to train sub-network $F_{m\to t}$ before cumulative training, which we name as CuDA-Net+.

The results in Table 1 show that although the backbone model DeepLab-v2 performs not well as RefineNet, our proposed method CuDA-Net (using DeepLab-v2 as the backbone) achieves a top performance, outperforming all state-of-the-art methods. We also achieve SOTA on ACDC [29] (ACDC-fog benchmark). Besides, we can see that the DA-based methods, which directly adapt the segmentation model from domain s to domain t, can not significantly improve the performance compared to our method. This is consistent with our assumption that general domain

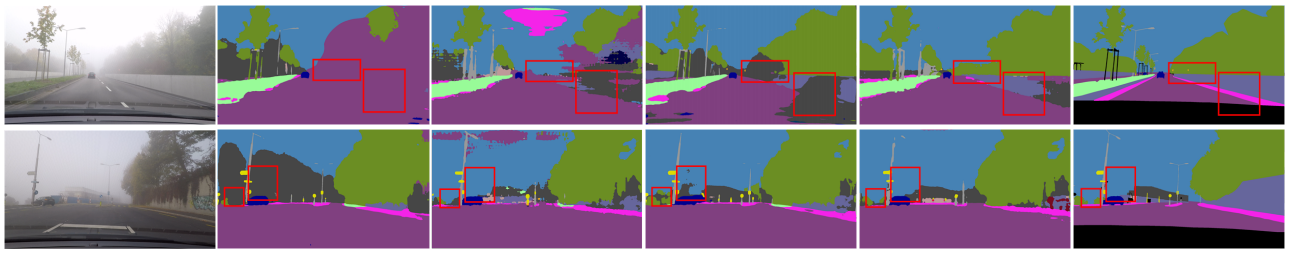

Figure 4. **The qualitative comparison with the SOTA methods**. The input images are randomly selected from Foggy Zurich-test. The red boxes clearly show that our method can better deal with the details than the SOTA methods.

Table 3. **Ablation study**. We conduct these experiments on Foggy Zurich-test dataset.

	Components			mIoU	gain
Initialization	Deeplabv2			25.89	+0.00
Style and Fog Decomposition	$F_{s\to m}$	$F_{m\to t}$	$F_{s\to t}$	mIoU	gain
	✓			39.16	+13.27
	✓	✓		42.49	+16.60
			✓	40.21	+14.32
	✓	✓	✓	43.06	+17.17
Cyclical Training	$T=1$	$T=2$	$T=3$	mIoU	gain
	✓			45.32	+19.43
		✓		45.78	+19.89
			✓	45.45	+19.56
Cumulative Loss	$L1$	cosine	$L2$	mIoU	gain
	✓			47.64	+21.75
		✓		47.23	+21.34
			✓	48.21	+22.32

adaptation methods can not perform well when the domain gap is too large and affected by different factors (style and fog), also proving the necessity of investigating both style and fog factor in this setting. The results also show that the defogging-based methods cannot always obtain good performances. It is because defogging-based methods require pair-wise training data to remove the fog, but we can not obtain this kind of data in SFSU.

In Table 1, when we introduce the synthetic foggy scene datasets—Foggy Cityscapes DBF simulated in CMAda3+ to our method, our CuDA-Net+ further improve the performance, outperforming CMAda3+ by 2.3% on FZ (3.7% on FD). Note that, our CuDA-Net+ improves by 0.9% from CuDA-Net on FZ when we only introduce 498 dense synthetic foggy images, indicating synthesizing foggy images and our CuDA-Net can complement each other very well. However, combining DISE [4] with the defogging method MSCNN [24] or the fog synthesis method SFSU [28] cannot yield better performance than only using DISE [4].

The qualitative comparison is shown in Figure 4. The red boxes clearly show that our method CuDA-Net can better deal with the details than CMAda3+, especially for the classes in the boundary of sky and other objects.

Table 4. **Different selection schemes for constructing m domain.** We compare three selection schemes when using different number of images. We test the trained model on Foggy Zurich-test dataset.

# Selected Images	w/o m Domain	Random	CNN-based	Manual
198	40.2	41.9	46.9	47.3
248	40.2	42.4	47.7	48.1
298	40.2	42.8	48.1	48.4

3.3. Discussion

In this section, we conduct a series of ablation studies to validate the contributions of individual components to the final foggy scene understanding.

Effectiveness of style and fog decomposition. In Table 3, the non-adapted model Deeplabv2, which is also the backbone of our CuDA-Net, only gives 25.89 mIoU on FZ. When using '$F_{s\to m}$', the performance increases to 39.16, revealing that the style adaptation matters. When using '$F_{s\to m}$+$F_{m\to t}$', i.e. first conducting style adaptation and then fog adaptation, we bring +3.33 mIoU gain, additionally indicating that the fog adaptation matters. Note that using '$F_{s\to m}$+$F_{m\to t}$' is 2.28 higher than only using '$F_{s\to t}$', which demonstrates that directly transferring with style and fog is not as good as two step adaptation. Other than that, using '$F_{s\to m}$+$F_{m\to t}$+$F_{s\to t}$' boosts the performance to 43.06, showing the necessity of dual-factor adaptation.

Effectiveness of cyclical training. We investigate the importance of cyclical training without L_{cum}, *i.e.*using Equation (4). As shown in Table 3, when we set T as 2, cyclical training improves the performance by 2.72. When we set T as 1 or 3, both results are close to 45.78, which shows the performance is not sensitive to the selection of T.

Effectiveness of cumulative loss. We also investigate the effects of cumulative loss L_{cum} in Table 3. We fix the T as 2 and use different distance metrics to calculate the domain discrepancy between two domains in the cumulative training. We find the L2 distance attains top performance. We also show some subjective segmentation results in Figure 5. They clearly indicate that segmentation results go better as more components are used in CuDA-Net.

Effects of different selection schemes for constructing m domain dataset. For constructing Clear Zurich, we man-

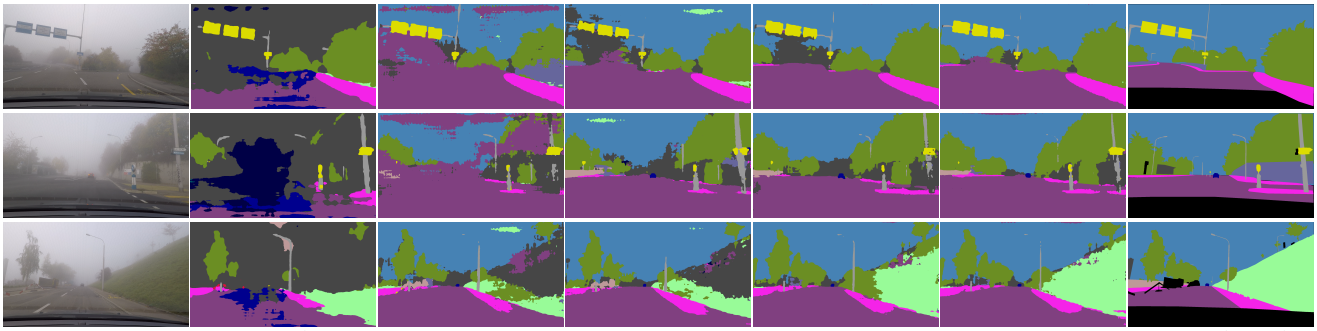

Figure 5. **Qualitative results of ablation study**. These experiments are conducted on the Foggy Zurich-test dataset. Each column shows the results of the proposed method with different components. The results show more clear spatial structure as more components are used.

Figure 6. **The ability of defogging**. We compare our defogged images generated by the $F_{m\to t}$ in CuDA-Net with those from the conventional defogging method GFN [25]. The input images are randomly selected from Foggy Zurich.

ually select 248 images from the light category of Foggy Zurich* [27] based on the human vision, to see whether they are clear or not. To prove its effectiveness, we also train a CNN to discriminate the clearness of the images in the light category of Foggy Zurich* [27] and select the top 248 images to construct the m domain. As shown in Table 4, we find the manual selection functions better than CNN-based selection, which shows the necessity of our manual selection scheme. Furthermore, when we randomly select images, the performance drops significantly, compared to CNN-based or manual selection, indicating a suitable criterion is necessary during the selection.

Visualization of Defogging. Although our CuDA-Net aims to transfer style and fog for foggy scene understanding, it is also capable of defogging foggy images during disentanglement learning, as mentioned in the cross-domain translation part of Section 2.1. In Figure 6, we visualize the results of defogging and compare our method with the defogging method GFN [25]. The results clearly show that our method can remove the fog well and does not destroy the content of the images, while GFN [25] brings in the color distortion.

Generalization to rainy and snowy scenes. Thanks to the ACDC [29] datasets, we can test our method on rainy and snowy scenes in Table 5. The results show that our proposed two-steps adaptation is better than directly adapting from the source domain to the target domain in other adverse scenes, indicating the potential of our method to address the understanding of different adverse scenes.

Table 5. **Generalization to rainy and snowy scenes.** We train our baseline on the ACDC rainy and snowy subsets and test it on the corresponding validation set, where $F_{s\to m+t}$ means we combine the m domain and t domain data as the whole target domain data.

Setting	$F_{s\to m+t}$	$F_{s\to m}$	$F_{s\to m}$+$F_{m\to t}$
ACDC (rain)	46.2	43.9	48.5
ACDC (snow)	44.8	42.6	47.2

4. Conclusion

In this paper, we propose the Cumulative style-fog-dual disentanglement Domain Adaptation method (CuDA-Net) for the SFSU task. We assume that the dual (style and fog) domain gap exists in SFSU, and that style, fog and dual factors have a cumulative relationship. Our method outperforms state-of-the-art methods on three widely-used datasets in SFSU and shows generalization ability to other adverse scenes, such as rainy and snowy scenes. We will make the code publicly available.

Limitation. 1. We chose DISE [4] as our baseline, which can be replaced with other new stronger disentanglement-based domain adaptation methods. By doing so, we believe our CuDA-Net can achieve better performance. **2.** We conduct primary experiments to showcase certain generalization ability to rainy and snowy scenes, and more detailed analysis can be done to verify whether the cumulative relationship exists in other adverse settings.

Acknowledgements. This work was supported by National Key R&D Project (2021YFC3320301) and National Natural Science Foundation of China (62171325). The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University. Zhixiang thanks the MEXT Scholarship and Value Exchange Engineering, a joint research project between Mercari, Inc. and RIISE.

References

[1] Naif Alshammari, Samet Akcay, and Toby P. Breckon. Competitive simplicity for multi-task learning for real-time foggy scene understanding via domain adaptation. *CoRR*, abs/2012.05304, 2020. 6

[2] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *CVPR*, 2021. 1, 6

[3] Dana Berman, Tali Treibitz, and Shai Avidan. Non-local image dehazing. In *CVPR*, 2016. 2, 6

[4] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *CVPR*, 2019. 1, 3, 6, 7, 8

[5] 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. *PAMI*, 2017. 6

[6] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, 2018. 1

[7] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, 2019. 1

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5

[9] Dengxin Dai, Christos Sakaridis, Simon Hecker, and Luc Van Gool. Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *IJCV*, 2020. 1, 2, 6

[10] Sébastien de Blois, Ihsen Hedhli, and Christian Gagné. Learning of image dehazing models for segmentation tasks. In *EUSIPCO*, 2019. 2

[11] Li Gao, Jing Zhang, Lefei Zhang, and Dacheng Tao. Dsp: Dual soft-paste for unsupervised domain adaptive semantic segmentation. In *ACMMM*, 2021. 1

[12] Rui Gong, Dengxin Dai, Yuhua Chen, Wen Li, and Luc Van Gool. Analogical image translation for fog generation. In *AAAI*, volume 1, page 2, 2021. 2, 6

[13] Xiaoqing Guo, Chen Yang, Baopu Li, and Yixuan Yuan. Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation. In *CVPR*, 2021. 1, 6

[14] Martin Hahner, Dengxin Dai, Christos Sakaridis, Jan-Nico Zaech, and Luc Van Gool. Semantic understanding of foggy scenes with purely synthetic data. In *ITSC*, 2019. 1, 2

[15] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *PAMI*, 2010. 2, 6

[16] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 3, 6

[17] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *CVPR*, 2020. 1

[18] Divya Kothandaraman, Rohan Chandra, and Dinesh Manocha. Ss-sfda: Self-supervised source-free domain adaptation for road segmentation in hazardous environments. *arXiv preprint arXiv:2012.08939*, 2020. 1

[19] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *ECCV*, 2020. 1, 6

[20] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 6

[21] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020. 1

[22] Srinivasa G. Narasimhan and Shree K. Nayar. Contrast restoration of weather degraded images. *PAMI*, 2003. 1

[23] Yanting Pei, Yaping Huang, Qi Zou, Yuhang Lu, and Song Wang. Does haze removal help cnn-based image classification? *ECCV*, 2018. 2

[24] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, 2016. 2, 6, 7

[25] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *CVPR*, 2018. 2, 6, 8

[26] Wenqi Ren, Jingang Zhang, Xiangyu Xu, Lin Ma, Xiaochun Cao, Gaofeng Meng, and Wei Liu. Deep video dehazing with semantic segmentation. *TIP*, 2018. 2

[27] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *ECCV*, 2018. 1, 2, 5, 6, 8

[28] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 2018. 1, 2, 5, 6, 7

[29] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 6, 8

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[31] Robby T Tan. Visibility in bad weather from a single image. In *CVPR*, 2008. 1

[32] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *WACV*, 2021. 1, 6

[33] Thanh-Dat Truong, Chi Nhan Duong, Ngan Le, Son Lam Phung, Chase Rainwater, and Khoa Luu. Bimal: Bijective maximum likelihood approach to domain adaptation in semantic scene segmentation. In *ICCV*, 2021. 1

[34] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 6

[35] 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. 1, 6

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

[37] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020. 1

[38] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *CVPR*, 2018. 2, 6

[39] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021. 1, 6

[40] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021. 1

[41] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, 2018. 1

[42] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *IJCV*, 2020. 2

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 6