Computational Visual Media
https://doi.org/10.1007/s41095-023-0362-4

Research Article

Class-conditional domain adaptation for semantic segmentation

Yue Wang[1], Yuke Li[2], James H. Elder[3], Runmin Wu[4], and Huchuan Lu[1] (✉)

Abstract Semantic segmentation is an important sub-task for many applications. However, pixel-level ground-truth labeling is costly, and there is a tendency to overfit to training data, thereby limiting the generalization ability. Unsupervised domain adaptation can potentially address these problems by allowing systems trained on labelled datasets from the source domain (including less expensive synthetic domain) to be adapted to a novel target domain. The conventional approach involves automatic extraction and alignment of the representations of source and target domains globally. One limitation of this approach is that it tends to neglect the differences between classes: representations of certain classes can be more easily extracted and aligned between the source and target domains than others, limiting the adaptation over all classes. Here, we address this problem by introducing a Class-Conditional Domain Adaptation (CCDA) method. This incorporates a class-conditional multi-scale discriminator and class-conditional losses for both segmentation and adaptation. Together, they measure the segmentation, shift the domain in a class-conditional manner, and equalize the loss over classes. Experimental results demonstrate that the performance of our CCDA method matches, and in some cases, surpasses that of state-of-the-art methods.

Keywords domain adaptation; generative adversarial networks; semantic segmentation; cityscapes

1 Introduction

Semantic segmentation is an important visual scene-understanding task with a wide range of applications, particularly in autonomous and assisted vehicle systems [1]. Recent deep network approaches (e.g., Refs. [2–4]) have achieved impressive results, but they require large training datasets with precise pixel-level ground-truth annotations. This may also lead to poor generalization ability due to the large domain shifts in appearance, viewpoint, and lighting between the source training and target testing domains [5].

These issues can potentially be addressed using the unsupervised domain adaptation method that attempts to identify and correct for a shift in the appearance of visual input between different domains. This is achieved by training a semantic segmentation model with a large number of synthetic source domain images that do not perfectly represent the appearance of real scenes but have easily obtainable ground-truth labels, as well as real-world target domain images whose ground-truth labels remain unknown. Therefore, a successful domain adaptation method will not only improve generalization but also avoid the time-consuming annotation for pixel-level multi-class segmentation of real-world scenes.

A common approach to solve the “domain shift” problem for deep network systems is to modify the weights of the network to render representations of target domain images more similar to the representations of source domain images. By minimizing the distance between the distributions of certain representations in both domains, a well-generalized model can be obtained. Some existing works have focused on representations in the

1 School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China. E-mail: Y. Wang, ellabear@mail.dlut.edu.cn; H. Lu, lhchuan@dlut.edu.cn (✉).

2 School of Computer Science, Wuhan University, Wuhan 430072, China. E-mail: sunfreshing@whu.edu.cn.

3 Department of Electrical Engineering and Computer Science, York University, Toronto M3J 1P3, Canada. E-mail: jelder@yorku.ca.

4 Department of Computer Science, the University of Hong Kong, Hong Kong 999077, China. E-mail: rmwu@cs.hku.hk.

Manuscript received: 2023-02-10; accepted: 2023-06-23

prediction space [6, 7], while others have focused on representations in feature (latent) space [8, 9]. Representational dissimilarity can be assessed using correlation distances [10] or maximum mean discrepancy [11]. However, recent studies have focused on generative adversarial methods [12] for unsupervised domain adaptation. This adversarial principle has become prominent since it achieved promising results in pixel-level prediction tasks [6, 13].

One limitation of previous studies on unsupervised domain adaptation for semantic segmentation is that they tend to measure feature extraction and alignment globally while ignoring the influence of different classes [14, 15]. The representation extraction and alignment ability of different classes can be affected by the occurrence frequency or appearance similarity between domains. An underlying tendency can be observed in that representations on classes with higher frequency can be easily extracted for segmentation, and representations on classes with higher appearance similarity between domains can be easily adapted. Therefore, the network may fail to extract meaningful feature representations from some classes using global segmentation prediction measurement. In addition, the global alignment of representations may cause the representations of some classes not to be fully adapted during training or cause the representations of classes that are already easily aligned to be mapped to incorrect classes.

To address the above issues, we propose a novel Class-Conditional Domain Adaptation (CCDA) method, which considers both adaptation and segmentation in a class-conditional manner. It comprises a class-conditional multi-scale discriminator and class-conditional loss functions for both segmentation and adaptation. Our class-conditional multi-scale discriminator encourages the network to align feature-level representations in a class-wise manner on both fine (pixel-level) and coarse (patch-level) spatial scales. For the coarse-scale branch, class-conditional adaptation is considered flexibly by requiring the discriminator to retain semantic information within each patch. It allows the adaptation on each class to be measured separately without neglecting any class. For the fine-scale branch, the class-conditional adaptation loss is equalized over classes to ensure that equal attention is paid to the alignment of each class. Moreover, the design of class-conditional segmentation loss function assists the network to fairly evaluate the segmentation performance on each class.

In summary, our proposed CCDA approach comprises three novel contributions:

- We propose a novel class-conditional multi-scale discriminator, which allows adaptation to be learnt in a class-wise manner.
- By equalizing class-conditional losses over classes for both segmentation and adaptation, the CCDA system pays equal attention to different classes.
- Experimental results demonstrate that the observed performance matches, and in some cases, surpasses that of state-of-the-art algorithms on several domain adaptation scenarios.

2 Related work

Domain adaptation: Research on domain adaptation for image classification has been conducted for many years, with a focus on solving the "domain shift" problem between different datasets on image-level representations. In the early stages, traditional distance minimization methods were proposed to reduce the distance between image representations from the source and target domains. For example, Ref. [16] used the maximum mean discrepancy (MMD) loss, and Ref. [17] applied coral loss. With the development of generative adversarial networks, many recent studies have achieved domain adaptation by minimizing the distance between representations using generative adversarial methods, which achieve better performance [18, 19].

Domain adaptation for semantic segmentation: Although substantial progress has been made in domain adaptation for image classification, pixel-level tasks are more challenging because of their direct dependence on local appearance. Nevertheless, increasing activity in autonomous vehicle applications has driven interest in domain adaptation for pixel-level segmentation of road scenes [8, 20]. Currently, the most popular approach to domain adaptation for pixel-level segmentation relies on adversarial learning, which is widely used for image generation [12, 21] and translation [22–24].

For domain adaptation, adversarial learning employs a discriminator on the segmentation network to align the source and target representations at either the prediction level [6, 14] or feature level [8, 9, 20]. Tsai et al. [6] employed an adversarial network to align pixel-level representations for adaptation. Vu et al. [14] then employed an indirect entropy minimization technique to improve the prediction-level adaptation. Luo et al. [9] used an information bottleneck to help remove task-independent information from feature-level representations during adaptation. Shan et al. [15] fused multi-level features for both segmentation and adaptation to allow gradients to flow into low-level CNN layers along a shorter path. Zhou et al. [13] performed domain adaptation on the affinity relationship between adjacent pixels to leverage the co-occurring patterns during adaptation.

In addition, the self-training approach can select pseudo ground-truth labels for target domain images to help supervise adaptation and improve performance [25–28]. The source-free method [29, 30] focuses on adaptation using only a well-trained source model and unlabeled target domain data. Additional techniques, such as image translation, can be combined with representation adaptation methods. Image translation focuses on narrowing down the domain shift between the source and target domains at the input image level by generating translated source domain images with target styles [27, 31]. However, most existing domain adaptation methods for semantic segmentation tended to measure segmentation and adaptation globally, while ignoring the influence of different classes, which may affect their performance.

Region-wise/class-wise domain adaptation: Adversarial approaches like Refs. [32, 33] tend to boost the domain adaptation performance for different classes or regions of the image. This suggests that a region- or class-wise domain adaptation approach is required to achieve good adaptation across all classes. Luo et al. [32] applied a co-training strategy to increase the weight of adaptation for poorly-aligned regions with inconsistent semantic predictions. Yang et al. [34] iteratively perturbed the intermediate feature maps with several attack objectives, which helps treat the information at each position evenly during adaptation. Tsai et al. [7] clustered patches based on spatial patterns and used cluster information as a guide to achieve better adaptation for each patch. However, these methods are still unable to equally and separately measure the segmentation and adaptation of each class, which may still limit their performance.

To achieve explicit class-wise adaptation, Chen et al. [8] applied 19 sub-discriminators during training, where each sub-discriminator is specially trained to measure the alignment of one class. Du et al. [33] further improved this framework by separating an entire feature into 19 sub-features based on the pseudo class label and inputting each sub-feature into the corresponding sub-discriminator for independent class-wise adaptation. Because the memory of the sub-discriminators varies linearly with the number of classes, it is less efficient to apply 19 sub-discriminators during training and may not be flexible when applied to datasets with more classes.

Here, we propose our CCDA method, which is a more holistic solution that entails one class-conditional multi-scale discriminator and class-conditional loss functions for both segmentation and adaptation. Using the class-conditional multi-scale discriminator, we allow the adaptation to be learnt in a class-wise manner. Equalizing the loss over classes for both segmentation and adaptation also helps pay equal attention to all classes. Meanwhile, by forcing the discriminator to maintain the semantic information while adversarially aligning the distributions between domains for each class, we can avoid using multiple sub-discriminators that apply one sub-discriminator to each class. Compared with the framework proposed by Du et al., our method is more efficient and flexible because we only require a single discriminator and still manage to separately and equally measure the adaptation for each class.

3 Methods

First, we describe the basic domain adaptation framework for pixel-level semantic segmentation. Next, we explain the innovations of our CCDA system in detail. It comprises two major components: class-conditional domain adaptation and class-conditional segmentation. In Section 3.2, we describe our class-conditional multi-scale discriminator, which contains both fine- and coarse-scale branches. In Section 3.3, we describe our class-conditional segmentation part. Figure 1 shows an overview of our CCDA system.

3.1 Basic domain adaptation architecture

We employed an adversarial learning approach to achieve unsupervised domain adaptation for semantic segmentation. The basic structure of this approach

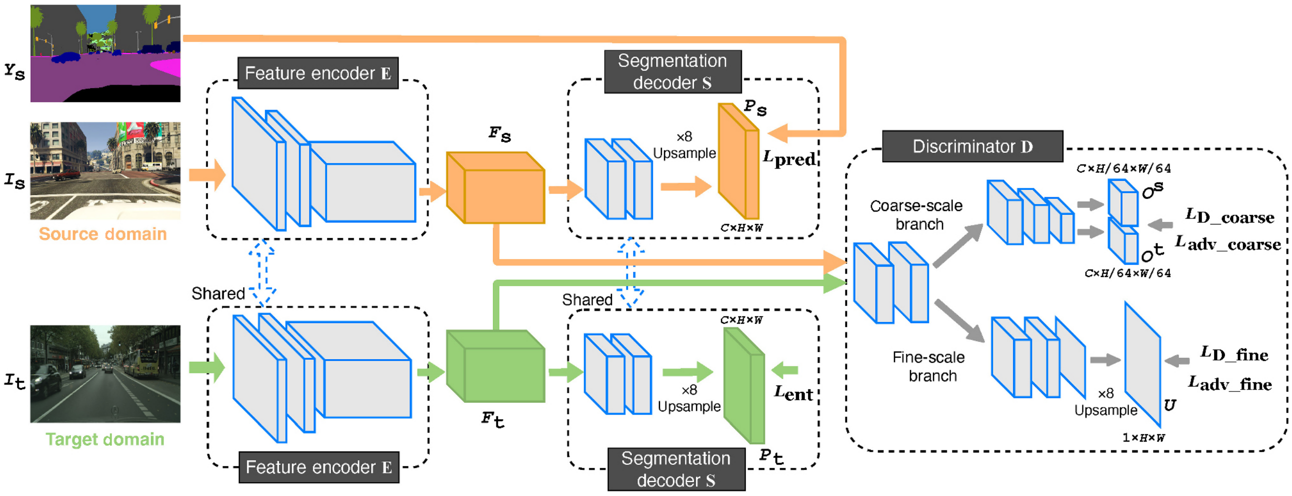


Fig. 1 Overview of our proposed Class-Conditional Domain Adaptation system. It consists of three parts: Feature Encoder **E**, Segmentation Decoder **S**, and Discriminator **D**. Orange arrows indicate the flow for the source domain, green arrows indicate the flow for the target domain, and grey arrows represent the flow for both domains. Given a source image I_s and target image I_t, we first pass them through **E** and **S** to obtain their feature-level representations F_s, F_t and pixel-level segmentation predictions P_s, P_t. Then, F_s and F_t from the two domains are input into **D** for feature-level representation alignment. To fairly measure the segmentation prediction for each class, we propose a class-conditional segmentation loss to supervise P_s of I_s based on its ground-truth label Y_s. To measure the feature alignment in a class-wise manner, we designed a class-conditional multi-scale discriminator. The fine-scale branch of **D** uses class-conditional adaptation loss to pay equal attention to the pixel-level alignment for each class, while the coarse-scale branch allows patch-level adaptation on each class to be separately and flexibly measured. More details of the coarse-scale branch in **D** are shown in Fig. 2.

typically consists of three modules: a feature encoder **E**, segmentation decoder **S**, and discriminator **D**. The image data consist of source and target data. Each source image $I_s \in \mathbb{R}^{3\times H\times W}$ is paired with ground-truth pixel-level segmentation label $Y_s \in \mathbb{R}^{C\times H\times W}$, H, W are the height and width of the image, respectively, and $C = 19$ denotes the number of semantic classes. The target image $I_t \in \mathbb{R}^{3\times H\times W}$ is assumed to have no ground-truth data available for training.

Our goal is to train the feature encoder **E** and segmentation decoder **S** to output a good pixel-level segmentation prediction $P_t \in \mathbb{R}^{C\times H\times W}$ for the target domain image. This is achieved through two processes: training **E** and **S** to output a good segmentation prediction $P_s \in \mathbb{R}^{C\times H\times W}$ for the source image I_s with the associated label Y_s, and using the discriminator **D** to align the feature-level representations F_s and F_t output by the feature encoder **E** for the two domains.

The first process (segmentation) is trained by minimizing the segmentation cross-entropy loss as Eq. (1):

$$\mathcal{L}_{\mathrm{seg}} = -\frac{1}{N}\sum_{h,w}\sum_{c} Y_{\mathrm{s}}[c,h,w]\log(P_{\mathrm{s}}[c,h,w]) \qquad (1)$$

where (h, w) denotes pixel position, and $c \in \{1, 2, \cdots, C\}$ represents a semantic class. $N = H \times W$ denotes the number of pixels. $Y_s[c, h, w]$ and $P_s[c, h, w]$ are the ground-truth and predicted state for class c at pixel (h, w). $P_s = \mathbf{S}(F_s) = \mathbf{S}(\mathbf{E}(I_s))$ is the output of the segmentation decoder **S**.

The second process (alignment) is trained adversarially to generate domain-invariant features. The discriminator module **D** attempts to distinguish feature representations from the source and target domains, minimizing

$$\mathcal{L}_{\mathrm{D1}} = \lambda_{\mathrm{sd}}\mathcal{L}_{\mathrm{bce}}(\mathbf{D}(F_{\mathrm{s}}), 0) + \lambda_{\mathrm{td}}\mathcal{L}_{\mathrm{bce}}(\mathbf{D}(F_{\mathrm{t}}), 1) \quad (2)$$

where $\mathcal{L}_{\mathrm{bce}}$ is the binary cross-entropy domain classification loss. The output channel of this basic discriminator **D** is 1 because it is for two classes (source and target domains). The source and target domain samples are assigned labels of 0 and 1, respectively. Normally, $\mathbf{D}(F)$ outputs a prediction that retains the resolution of the input feature representation instead of a prediction with one single value at the global image level. Therefore, adaptation can be measured more precisely by calculating the average loss over all positions in the input feature. Concurrently, the feature encoder **E** attempts to confuse **D**, minimizing

$$\mathcal{L}_{\mathrm{adv1}} = \lambda_{\mathrm{sa}}\mathcal{L}_{\mathrm{bce}}(\mathbf{D}(F_{\mathrm{s}}), 1) + \lambda_{\mathrm{ta}}\mathcal{L}_{\mathrm{bce}}(\mathbf{D}(F_{\mathrm{t}}), 0) \quad (3)$$

This basic structure extracts the semantic representations and produces an alignment of the features globally among all classes. It does not consider that different classes may have different influences on the segmentation and adaptation. This tends to cause predictions on some lower-frequency classes that do not contribute substantially to the cross-entropy loss, or representations on some classes that are not fully adapted owing to the dissimilar appearance between domains. This could also deconstruct the existing alignment and cause regions that belong to classes that have already been well adapted to be mistakenly adapted to other classes. Meanwhile, because the feature map computed by the feature encoder **E** is spatiotopic but its resolution is reduced relative to the input image, the alignment achieved by this process is at a specific intermediate scale of the feature map, which may not capture the domain shift at smaller or larger scales. These observations motivate our class-conditional multi-scale discriminator and class-conditional segmentation.

3.2 Class-conditional multi-scale discriminator

Our proposed class-conditional multi-scale discriminator is composed of fine- and coarse-scale branches (Fig. 1). The fine-scale branch measures alignment at the pixel level based on the basic architecture with modified loss functions. It captures spatially-detailed domain shift phenomena in a class-wise manner. The coarse-scale branch measures the class-conditional alignment at the patch level, which is coarser than the feature scale with equal class information. First, we describe how to perform this class conditioning by explaining the design of the coarse-scale class label. Then, we elaborate on the structure of the class-conditional coarse-scale discriminator branch as well as the fine-scale branch.

3.2.1 Coarse-scale class label

We define a coarse-scale binary class label W with length C that indicates the presence or absence of each class within a rectangular patch of the image. It should be noted that a patch may contain multiple classes. For the source image, W_{s} is computed by analyzing the pixel-level ground-truth label Y_{s} within the image back-projection of a patch. For a patch at position (j,k), if any pixel within the back-projected region of the image has class c, we set $W_{\mathrm{s}}[c,j,k]=1$; otherwise, we set $W_{\mathrm{s}}[c,j,k]=0$.

For the target domain image, we do not have ground-truth label. Instead, we assign the coarse-scale class label based on the projected pixel-level prediction P_{t} of our segmentation module **S** for the patch. In particular, for the patch at position (j,k), given a confidence threshold th_w, if any pixel (h,w) within the back-projected region of the image contains $P_{\mathrm{t}}[c,h,w]>th_w$, we set $W_{\mathrm{t}}[c,j,k]=1$; otherwise, we set $W_{\mathrm{t}}[c,j,k]=0$.

Note that binarizing the patch-based class label W equalizes the class information at the patch level: $W[c,j,k]=1$ if the patch (j,k) contains any pixels of class c, regardless of the number. This has the benefit of maintaining semantic information in our discriminator without neglecting any classes in a patch. It also applies equal attention to all the classes that a patch contains and boosts the adaptation performance in a class-wise manner.

3.2.2 Class-conditional coarse-scale branch

In standard feature-level domain adaptation, the discriminator output for each patch indicates the domain of the entire patch (in our case, 0 for source domain, 1 for target domain). To apply class-conditional adaptation, Du et al. designed a sub-discriminator system with 19 sub-discriminators, each specially trained for one corresponding class, and achieved good performance. By contrast, the class-conditional discriminator we designed consists of both semantic classification and adaptation. It maintains semantic information while measuring the class-wise adaptation adversarially, which avoids the usage of sub-discriminators. The output of our class-conditional coarse-scale discriminator branch consists of two vectors, O^{s} and O^{t}, each of length C. $O^{\mathrm{s}}[c,j,k]$ estimates the probability that the patch (j,k) contains one or more pixels drawn from class c of the source domain, while $O^{\mathrm{t}}[c,j,k]$ estimates the probability that the patch (j,k) contains one or more pixels drawn from class c of the target domain.

The advantage of this two-vector representation is that it allows us to multiplex both domain and class information, informing both adversarial adaptation loss based on the class and non-adversarial classification loss (Fig. 2). In particular, to determine the non-adversarial classification loss, we form the vector $O^c=\sigma\left(O^{\mathrm{s}}+O^{\mathrm{t}}\right)$, where the sigmoid function $\sigma(\cdot)$ is applied separately for each class. Therefore, $O^c[c,j,k]$ estimates the probability that the patch (j,k) contains one or more pixels drawn from class c. We calculate the classification loss

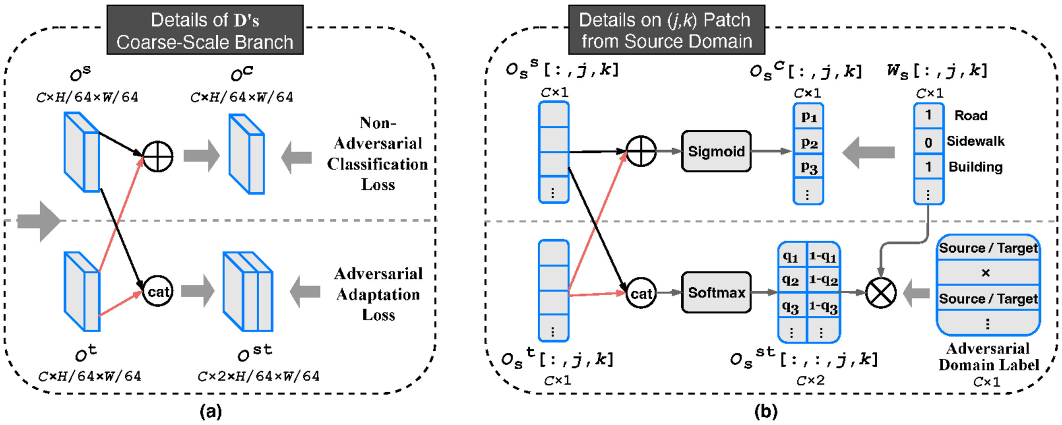


Fig. 2 Details of our class-conditional discriminator on the coarse-scale branch. (a) Overview. (b) Details for one patch (j, k) from source domain as an example. For better understanding, $p_c = O^c_{\mathrm{s}}[c, j, k]$ is the estimated probability that the patch (j, k) from the source domain contains one or more pixels belonging to class c. $q_c = O^{\mathrm{st}}_{\mathrm{s}}[c, 1, j, k]$ represents the estimated probability that pixels of class c in patch (j, k) belong to the source domain. $O^{\mathrm{st}}_{\mathrm{s}}[c, 2, j, k] = 1 - q_c$ indicates the probability that pixels of class c in patch (j, k) belong to the target domain. Here, the usage of class-conditional domain vectors $O^{\mathrm{s}}_{\mathrm{s}}$ and $O^{\mathrm{t}}_{\mathrm{s}}$ (O^{s} and O^{t} for source domain) allows multiplexing of both domain and class information, informing an adversarial adaptation loss based on class and a non-adversarial classification loss. O^c_{s} is supervised by the corresponding class label W_{s} for maintaining semantic information, while $O^{\mathrm{st}}_{\mathrm{s}}$ is supervised by domain label for class-wise adaptation with W_{s} as weights. Therefore, it allows a flexible and separate feature alignment for each class.

for both domains using a binary cross-entropy loss $\mathcal{L}_{\mathrm{bce}}(O^c, W)$ averaged over all classes, because each patch can contain multiple classes. Including this classification loss in the discriminator encourages the feature-level domain alignment to preserve the segmentation class information for patches from both the source and target domain images. Because of the binary nature of the coarse-scale class label vector W, we prevent the discriminator from neglecting any classes with a small number of pixels in a patch.

To obtain the adversarial adaptation loss, we form a $C \times 2$ matrix $O^{\mathrm{st}} = f\left([O^{\mathrm{s}}, O^{\mathrm{t}}]\right)$ for each patch, where $f(\cdot)$ is the softmax operation over rows (with a channel number of 2, where the first $C\times 1$ is for source, and the second is for target). Therefore, for patch (j, k), $O^{\mathrm{st}}[c, 1, j, k]$ represents the probability that the pixels of class c in this patch belong to the source domain, and $O^{\mathrm{st}}[c, 2, j, k]$ represents the probability that the pixels of class c in this patch belong to target domain, $O^{\mathrm{st}}[c, 1, j, k] + O^{\mathrm{st}}[c, 2, j, k] = 1$. O^{st} indicates that for each class, the probability that any pixels of this class present in a patch are drawn from the source versus target domains. Therefore, it allows the adaptation of each class that occurs within the patch to be measured separately with only one discriminator.

To form the final class-wise discriminator domain adaptation loss for the coarse-scale branch, we average the class-conditional loss over the classes present in a patch. This means weighting the sum of the losses using ground-truth patch-level class label W_{s} for the source domain and predicted patch-level class prediction O^c_{t} (O^c for the target sample) for the target domain, and then dividing by the sum of the weights. Combined with the non-adversarial classification loss, the total patch-level discriminator loss is

$$\begin{aligned}\mathcal{L}_{\mathrm{D\_coarse}} = {} & \mathcal{L}_{\mathrm{bce}}(O^c_{\mathrm{s}}, W_{\mathrm{s}}) + \mathcal{L}_{\mathrm{bce}}(O^c_{\mathrm{t}}, W_{\mathrm{t}}) \\ & - \frac{\lambda_{\mathrm{sd}}}{M}\sum_{j,k}\frac{\sum_c W_{\mathrm{s}}[c,j,k]\log(O^{\mathrm{st}}_{\mathrm{s}}[c,1,j,k])}{\sum_c W_{\mathrm{s}}[c,j,k]} \\ & - \frac{\lambda_{\mathrm{td}}}{M}\sum_{j,k}\frac{\sum_c O^c_{\mathrm{t}}[c,j,k]\log(O^{\mathrm{st}}_{\mathrm{t}}[c,2,j,k])}{\sum_c O^c_{\mathrm{t}}[c,j,k]}\end{aligned} \tag{4}$$

where $O^{\mathrm{st}}_{\mathrm{s}}$ is the output O^{st} for the source domain image, $O^{\mathrm{st}}_{\mathrm{t}}$ is the output O^{st} for the target domain image, and M is the number of patches. Here, the discriminator is trained to precisely distinguish the features from both domains for each class. Therefore, we expect it to predict $O^{\mathrm{st}}_{\mathrm{s}}[c, 1, j, k] = 1$ for the source domain and $O^{\mathrm{st}}_{\mathrm{t}}[c, 2, j, k] = 1$ for the target domain.

The generative component of the adversarial loss is symmetrically defined as

$$\begin{aligned}\mathcal{L}_{\mathrm{adv\_coarse}} = {} & \mathcal{L}_{\mathrm{bce}}(O^c_{\mathrm{s}}, W_{\mathrm{s}}) + \mathcal{L}_{\mathrm{bce}}(O^c_{\mathrm{t}}, W_{\mathrm{t}}) \\ & - \frac{\lambda_{\mathrm{sa}}}{M}\sum_{j,k}\frac{\sum_c W_{\mathrm{s}}[c,j,k]\log(O^{\mathrm{st}}_{\mathrm{s}}[c,2,j,k])}{\sum_c W_{\mathrm{s}}[c,j,k]} \\ & - \frac{\lambda_{\mathrm{ta}}}{M}\sum_{j,k}\frac{\sum_c O^c_{\mathrm{t}}[c,j,k]\log(O^{\mathrm{st}}_{\mathrm{t}}[c,1,j,k])}{\sum_c O^c_{\mathrm{t}}[c,j,k]}\end{aligned} \tag{5}$$

Here, we expect it to predict $O^{\mathrm{st}}_{\mathrm{s}}[c, 2, j, k] = 1$ for the source domain and $O_\text{t}^\text{st}[c,1,j,k] = 1$ for the target domain to confuse the discriminator.

3.2.3 Class-conditional fine-scale discriminator

Coarse-scale class-conditional adaptation can capture larger-scale domain shift effects but may not capture shifts in finer detail. Thus, we employ a class-conditional fine-scale discriminator operating at the pixel level. The scale of the feature representations in this fine-scale discriminator branch remains, and the output is upsampled to produce a fine-scale domain classification $U_\text{s} \in \mathbb{R}^{1\times H\times W}$ for I_s and $U_\text{t} \in \mathbb{R}^{1\times H\times W}$ for I_t that match the original input size. For this fine-scale discriminator branch, we do not need to retain semantic information, but we still evaluate the performance of the adaptation in a class-wise manner using a designed class-conditional loss that equalizes the performance among all classes.

For the source domain, we employ the ground-truth class label Y_s to calculate the loss for each class and equally average over the classes to form a class-conditional binary cross-entropy loss:

$$\mathcal{L}_{\text{cbce\_s}}(U_\text{s}, Y_\text{s}, l_\text{d}) = \frac{1}{C^*}\sum_c \frac{\sum_{h,w} Y_\text{s}[c,h,w]\mathcal{L}_\text{bce}(U_\text{s}[h,w], l_\text{d})}{\sum_{h,w} Y_\text{s}[c,h,w]+\epsilon} \tag{6}$$

where C^* is the number of classes present in the source image. The ground-truth domain label l_d is set to $l_\text{d} = 0$ when training the discriminator $\mathbf{D}$ and $l_\text{d} = 1$ when training the encoder $\mathbf{E}$ and segmentation decoder $\mathbf{S}$, to confuse the discriminator. ϵ is a small constant that prevents division by 0 for the classes that do not appear in the ground truth within an image.

For the target domain image, we do not have a ground-truth class label; therefore, we employ pixel-level class prediction P_t instead to form a pseudo label $\hat{Y}_\text{t}$ by selecting the class with the highest prediction value:

$$\hat{Y}_\text{t}[c,h,w] = \begin{cases} 1, & \text{if } \ c = \arg\max_c P_\text{t}[c,h,w] \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

For some pixels, P_t may have low entropy, which means that the network is confident and the pseudo labels on these pixels may be a good estimate of the ground truth class. For other pixels, P_t may have high entropy, which can be considered as a sign that the domain shift may be interfering with classification. Thus, the adaptation loss for pixels with uncertain predictions can be upweighted to improve adaptation to the domain shift. In particular, we designate these ambiguous pixels using the label $A_\text{t} \in \mathbb{R}^{1\times H\times W}$:

$$A_\text{t}[h,w] = \begin{cases} 1, & \text{if } \ \max_c P_\text{t}[c,h,w] < th_\text{a} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where th_a is a threshold constant for selecting the uncertain pixels. We then add a term to the fine-scale domain adaptation loss that serves to upweight these regions during feature alignment.

Thus, the final class-conditional binary cross-entropy loss for the target domain images becomes

$$\begin{aligned} &\mathcal{L}_{\text{cbce\_t}}(U_\text{t}, \hat{Y}_\text{t}, l_\text{d}) = \\ &\frac{1}{C^*+1}\Bigg(\sum_c \frac{\sum_{h,w} \hat{Y}_\text{t}[c,h,w]\mathcal{L}_\text{bce}(U_\text{t}[h,w], l_\text{d})}{\sum_{h,w} \hat{Y}_\text{t}[c,h,w]+\epsilon} \\ &\qquad + \lambda_n \frac{\sum_{h,w} A_\text{t}[h,w]\mathcal{L}_\text{bce}(U_\text{t}[h,w], l_\text{d})}{\sum_{h,w} A_\text{t}[h,w]+\epsilon}\Bigg) \end{aligned} \tag{9}$$

where C^* denotes the number of classes present in the target domain image, as predicted by $\hat{Y}_\text{t}$. It uses $l_\text{d} = 1$ to train $\mathbf{D}$ and $l_\text{d} = 0$ to train $\mathbf{E}$ and $\mathbf{S}$, to confuse the discriminator.

The fine-scale class-conditional discriminator loss for both the source and target domain images is then

$$\mathcal{L}_\text{D2} = \lambda_\text{sd}\mathcal{L}_{\text{cbce\_s}}(U_\text{s}, Y_\text{s}, 0) + \lambda_\text{td}\mathcal{L}_{\text{cbce\_t}}(U_\text{t}, \hat{Y}_\text{t}, 1) \tag{10}$$

The generative component of the adversarial fine-scale loss trained on feature encoder $\mathbf{E}$ and segmentation decoder $\mathbf{S}$ is symmetrically defined as

$$\mathcal{L}_\text{adv2} = \lambda_\text{sa}\mathcal{L}_{\text{cbce\_s}}(U_\text{s}, Y_\text{s}, 1) + \lambda_\text{ta}\mathcal{L}_{\text{cbce\_t}}(U_\text{t}, \hat{Y}_\text{t}, 0) \tag{11}$$

For stability, we blend these class-conditional fine-scale losses with the conventional losses defined in Eqs. (2) and (3) to obtain the adaptation loss for the fine-scale branch:

$$\mathcal{L}_{\text{D\_fine}} = \beta\mathcal{L}_\text{D1} + (1-\beta)\mathcal{L}_\text{D2} \tag{12}$$

$$\mathcal{L}_{\text{adv\_fine}} = \beta\mathcal{L}_\text{adv1} + (1-\beta)\mathcal{L}_\text{adv2} \tag{13}$$

where β is a weight to combine the losses.

Therefore, the overall discriminator and adversarial losses of our class-conditional multi-scale discriminator combine the losses from both fine-scale and coarse-scale branches:

$$\mathcal{L}_{\text{D\_all}} = \mathcal{L}_{\text{D\_fine}} + \mathcal{L}_{\text{D\_coarse}} \tag{14}$$

$$\mathcal{L}_{\text{adv\_all}} = \mathcal{L}_{\text{adv\_fine}} + \mathcal{L}_{\text{adv\_coarse}} \tag{15}$$

3.3 Class-conditional segmentation loss

The conventional loss employed for pixel-level semantic segmentation is pixel-level cross-entropy loss.

This means that the segmentation predictions are measured globally among the entire image, regardless of the class information. However, this method has a drawback in that some classes tend to be less frequent among the datasets or have objects with smaller sizes at the pixel level, which do not contribute substantially to the loss function. This conventional global segmentation loss has an additional consequence for the domain adaptation system in that the system may never learn how to align representations across domains for these classes.

Here, we introduce a modified class-conditional loss for segmentation that serves to distribute the loss more evenly across classes. Specifically, we employ the concept of dice loss [35] to train the segmentation network. Dice loss is widely used in medical image segmentation [36, 37] and has the form in Eq. (16):

$$\mathcal{L}_{\text{dice}} = 1 - \frac{1}{C}\sum_{c}\Big(\frac{2\sum_{h,w} Y_{\text{s}}[c,h,w]P_{\text{s}}[c,h,w]}{\sum_{h,w}(Y_{\text{s}}[c,h,w]+P_{\text{s}}[c,h,w])+\epsilon}\Big) \tag{16}$$

Note that the loss is similar in spirit to intersection-over-union and can equalize the contribution of each class to measure segmentation performance. ϵ is a small constant that prevents division by 0 for classes that do not appear in the ground-truth and prediction.

Dice loss can measure segmentation by class, which tends to increase the weight of loss in rare classes. However, this may also introduce instability during training. Therefore, we employ a combination of the dice loss and cross-entropy loss (Eq. (1)) to form our segmentation prediction loss:

$$\mathcal{L}_{\text{pred}} = \alpha\mathcal{L}_{\text{seg}} + (1-\alpha)\mathcal{L}_{\text{dice}} \tag{17}$$

where α is a weight to combine the losses.

3.4 Complete training loss

Following Refs. [14, 34, 38], we also add a regular entropy minimization loss $\mathcal{L}_{\text{ent}}$ for the segmentation prediction of the target domain image, which encourages our model to produce predictions with high confidence:

$$\mathcal{L}_{\text{ent}} = -\frac{\lambda_{\text{ent}}}{\log(C)N}\sum_{h,w}\sum_{c} P_{\text{t}}[c,h,w]\log(P_{\text{t}}[c,h,w]) \tag{18}$$

where λ_{ent} is a weight to balance the losses.

In summary, the complete training process combines class-conditional segmentation loss (Eq. (17)), class-conditional domain adaptation discriminator loss (Eq. (14)), adversarial loss (Eq. (15)), and entropy minimization loss (Eq. (18)):

$$\min_{\mathbf{D}} \mathcal{L}_{\text{D\_all}} \tag{19}$$

$$\min_{\mathbf{E},\mathbf{S}} \ \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{adv\_all}} + \mathcal{L}_{\text{ent}} \tag{20}$$

4 Experiments

4.1 Datasets and implementation details

Following most domain adaptation for segmentation methods, we evaluated our class-conditional domain adaptation method on semantic segmentation using two synthetic source domain datasets (GTA5 [39] and SYNTHIA [40]) and a real-world target domain dataset (Cityscapes [41]). This defines two adaptation tasks: GTA5 → Cityscapes and SYNTHIA → Cityscapes. The GTA5 dataset comprises 24,966 images, while the SYNTHIA dataset comprises 9400 images. Both synthetic datasets include pixel-level ground-truth semantic segmentation labels. The Cityscapes dataset contains 2975 training images and 500 validation images. To train the proposed domain adaptation model, we employed both images and ground-truth labels from either the GTA5 or SYNTHIA dataset as the source domain, and only the images (not the labels) from the Cityscapes training set as the target domain. We evaluated our model on the Cityscapes validation set over 19 classes for GTA5 → Cityscapes task and over 13 and 16 classes for SYNTHIA → Cityscapes task, as per convention [6, 26].

We implemented our training and evaluation in PyTorch on a single GeForce RTX 2080 Ti GPU with 11 GB of memory. We used the DeepLab-v2 [42] framework with a small pre-trained VGG16 [43] model as the backbone for our feature encoder **E** and segmentation decoder **S**. For the discriminator module **D**, the fine-scale branch has a structure similar to that in Ref. [6]. For the coarse-scale branch, we share the first two convolution layers with the fine-scale branch and then apply three convolution layers with channel number $\{256, 512, C \times 2\}$ and a kernel size of 3 and stride of 2 for downsampling. Except for the last convolution layers in both branches, each convolution layer in our discriminator module is followed by a Leaky-ReLU [44] layer with a slope of 0.2 for negative inputs. We also applied a two-stage training strategy using a self-training process, which is also used in Refs. [7, 25]. In stage 1, we trained an initial CCDA model first for 100k iterations, and in stage 2, we further fine-tuned our entire CCDA model with another 100k iterations as well as adding self-training on the target domain.

To train our feature encoder **E** and segmentation decoder **S**, we used a Stochastic Gradient Descent (SGD) optimizer [45] with a momentum of 0.9 and weight decay of 5×10^{-4}. The initial learning rate was set to 2.5×10^{-4} and decayed during training. For the discriminator module **D**, we applied the ADAM [46] optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.99$. The initial learning rate was set to 1×10^{-4} and decayed using the same policy as SGD. Our model was trained using batch size of two with one source domain image and one target domain image, and we resized the input images as $H \times W = 512 \times 1024$, which is the same as in Ref. [9]. Therefore, the number of patches for the coarse-scale discriminator branch $M = \dfrac{H}{64} \times \dfrac{W}{64} = 8 \times 16$. We set the thresholds $th_w = 0.4$, $th_{\rm a} = 0.95$. We also set $\lambda_{\rm sa} = \lambda_{\rm ta} = 0.0003$, $\lambda_{\rm sd} = \lambda_{\rm td} = 0.5$ across all loss functions, and $\alpha = 0.5$, $\beta = 0.4$, $\lambda_n = 3$, $\lambda_{\rm ent} = 0.05$ for the blend losses.

4.2 Comparison with state-of-the-art methods

Tables 1 and 2 summarize the performance of our overall CCDA method compared with state-of-the-art methods on the two transfer tasks GTA5 → Cityscapes and SYNTHIA → Cityscapes, respectively. For a fair comparison, we compared our method with the state-of-the-art methods using the same VGG16 backbone. These methods include adaptation on prediction-level representation methods (A-P): AdaptSeg [6], ADVENT [14], DPR [7], SSP [15], APO [34], ASA [13], TTDA [47]; adaptation on feature-level representation methods (A-F): FCNsW [20], Cross-city [8], SIBIN [9], OCE [38], SSF-DAN [33]; adaptation on both prediction- and feature-level representation method (A-PF): CLAN [32]; self-training (ST) methods: CBST-SP [25], CDA [26]; and source-free (SF) methods: SFDA [29], UBNA [30]. Here, almost all adaptation on representation methods apply adversarial learning as our method does, except for OCE. We also present visual comparisons in Fig. 3 and Fig. 4. However, because only a few state-of-the-art methods provide the available code and pretrained models, we could not obtain the predicted segmentation maps of all methods. Therefore, we only present visual comparisons with methods that have available code and pretrained models, as well as better performance, in these two figures.

Table 1 Adaptation from GTA5 to Cityscapes. We present the per-class and mean IoU. Here, "A" represents adaptation on representation methods, "-P" represents adaptation on prediction-level representation, and "-F" represents adaptation on feature-level representation. "ST" and "SF' represent self-training and source-free methods, respectively. We highlight the best and second-best results in each column in red and blue, respectively

		GTA5 → Cityscapes																			
	Meth.	road	side.	buil.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	mbike.	bike	mIoU
CDA	ST	72.9	30.0	74.9	12.1	13.2	15.3	16.8	14.1	79.3	14.5	75.5	35.7	10.0	62.1	20.6	19.0	0.0	19.3	12.0	31.4
CBST-SP		90.4	50.8	72.0	18.3	9.5	27.2	28.6	14.1	82.4	25.1	70.8	42.6	14.5	76.9	5.9	12.5	1.2	14.0	28.6	36.1
SFDA	SF	81.8	35.4	82.3	21.6	20.2	25.3	17.8	4.7	80.7	24.6	80.4	50.5	9.2	78.4	26.3	19.8	11.1	6.7	4.3	35.9
UBNA		79.9	29.9	78.1	21.1	16.5	33.8	29.7	20.6	75.6	18.4	78.0	58.4	14.6	79.4	14.8	13.0	5.8	14.6	10.6	36.5
AdaptSeg	A-P	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
ADVENT		86.9	28.7	78.7	28.5	25.2	17.1	20.3	10.9	80.0	26.4	70.2	47.1	8.4	81.5	26.0	17.2	18.9	11.7	1.6	36.1
SSP		87.7	30.8	78.5	23.2	20.3	25.8	24.5	14.1	80.2	30.1	73.6	48.9	11.8	82.2	24.1	22.5	0.7	13.7	1.4	36.5
ASA		86.9	32.5	79.0	22.8	23.1	20.7	22.0	12.6	80.0	32.2	68.5	43.6	11.9	81.3	20.8	9.6	4.2	16.9	8.5	35.6
DPR		87.3	35.7	79.5	32.0	14.5	21.5	24.8	13.7	80.4	32.0	70.5	50.5	16.9	81.0	20.8	28.1	4.1	15.5	4.1	37.5
APO		88.4	34.2	77.6	23.7	18.3	24.8	24.9	12.4	80.7	30.4	68.6	48.9	17.9	80.8	27.0	27.2	6.2	19.1	10.2	38.0
TTDA		88.7	38.6	80.2	26.0	21.5	22.3	25.0	14.7	83.2	32.3	77.0	53.0	17.5	81.1	21.3	21.5	0.0	21.5	7.8	38.6
CLAN	A-PF	90.4	40.2	80.6	25.1	21.8	27.6	24.2	19.6	83.1	33.9	74.1	47.7	9.5	83.9	27.0	27.1	3.4	17.5	0.9	38.8
FCNsW	A-F	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
SIBIN		83.4	13.0	77.8	20.4	17.5	24.6	22.8	9.6	81.3	29.6	77.3	42.7	10.9	76.0	22.8	17.9	5.7	14.2	2.0	34.2
OCE		86.0	13.5	79.4	20.4	18.5	21.5	27.6	15.2	80.8	21.9	72.6	46.3	18.1	80.0	16.9	13.1	1.0	14.6	2.0	34.2
SSF-DAN		88.7	32.1	79.5	29.9	22.0	23.8	21.7	10.7	80.8	29.8	72.5	49.5	16.1	82.1	23.2	18.1	3.5	24.4	8.1	37.7
Ours		91.1	45.1	81.1	29.8	23.2	30.1	31.2	19.7	81.3	29.4	74.5	54.0	15.6	81.4	21.3	20.3	3.8	21.7	6.0	40.0

Table 2 Adaptation from SYNTHIA to Cityscapes. The table is annotated the same as Table 1, while mIoU and mIoU* are averaged over 16 and 13 classes, respectively

SYNTHIA → Cityscapes																		
	Meth.	road	side.	buil.	wall*	fence*	pole*	light	sign	vege.	sky	pers.	rider	car	bus	mbike.	bike	mIoU	mIoU*
CDA	ST	57.4	23.1	74.7	0.5	0.6	14.0	5.3	4.3	77.8	73.7	45.0	11.0	44.8	21.2	1.9	20.3	29.7	35.4
CBST-SP		69.6	28.7	69.5	12.1	0.1	25.4	11.9	13.6	82.0	81.9	49.1	14.5	66.0	6.6	3.7	32.4	35.4	36.1
UBNA	SF	71.5	27.3	72.9	2.5	0.3	32.0	12.7	16.7	74.6	75.4	47.1	13.6	61.4	8.5	8.3	29.2	34.6	41.0
AdaptSeg	A-P	78.9	29.2	75.5	—	—	—	0.1	4.8	72.6	76.7	43.4	8.8	71.1	16.0	3.6	8.4	—	37.6
ADVENT		67.9	29.4	71.9	6.3	0.3	19.9	0.6	2.6	74.9	74.9	35.4	9.6	67.8	21.4	4.1	15.5	31.4	36.6
ASA		72.6	24.2	74.2	8.6	0.6	21.3	6.1	12.6	73.7	77.0	42.3	13.0	67.9	19.1	6.0	14.3	33.3	38.7
DPR		72.6	29.5	77.2	3.5	0.4	21.0	1.4	7.9	73.3	79.0	45.7	14.5	69.4	19.6	7.4	16.5	33.7	39.6
APO		82.9	31.4	72.1	—	—	—	10.4	9.7	75.0	76.3	48.5	15.5	70.3	11.3	1.2	29.4	—	41.1
TTDA		82.3	40.8	77.0	8.8	1.1	23.5	8.9	15.4	79.2	77.8	43.4	14.3	64.0	26.5	5.9	17.6	36.7	42.5
CLAN	A-PF	82.0	33.7	79.8	—	—	—	6.4	8.9	78.7	82.5	49.1	12.9	75.9	21.9	5.1	13.3	—	42.3
FCNsW	A-F	11.5	19.6	30.8	4.4	0.0	20.3	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	20.2	22.9
Cross-city		62.7	25.6	78.3	—	—	—	1.2	5.4	81.3	81.0	37.4	6.4	63.5	16.1	1.2	4.6	—	35.7
SIBIN		70.1	25.7	80.9	—	—	—	3.8	7.2	72.3	80.5	43.3	5.0	73.3	16.0	1.7	3.6	—	37.2
OCE		78.3	30.1	78.0	1.7	0.1	24.1	12.0	14.6	79.7	79.1	51.4	15.5	74.4	23.7	9.1	22.7	37.1	43.7
SSF-DAN		87.1	36.5	79.7	—	—	—	13.5	7.8	81.2	76.7	50.1	12.7	78.0	35.0	4.6	1.6	—	43.4
Ours		83.3	35.3	77.9	5.2	0.4	27.4	12.3	12.6	79.9	81.7	41.4	12.4	71.3	22.6	5.4	25.5	37.2	43.2

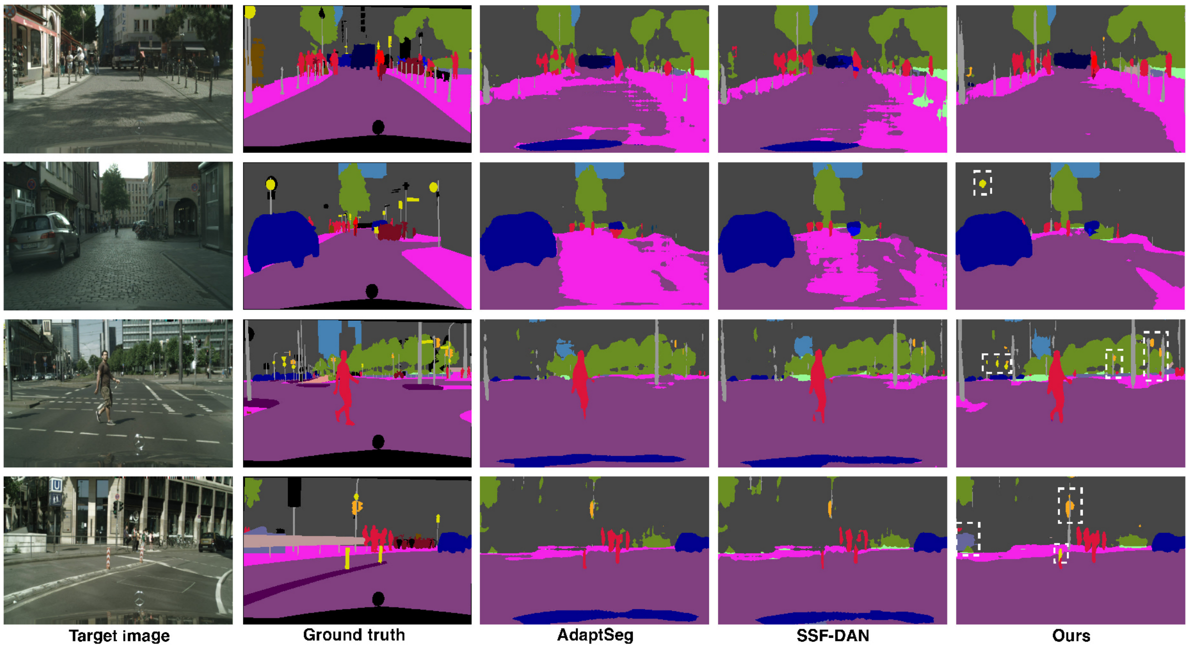


Fig. 3 Qualitative results of semantic segmentation on the GTA5 → Cityscapes task. For each target image, we show the corresponding ground-truth map and the results of AdaptSeg, SSF-DAN, and our proposed CCDA method. We highlight some improved predictions with white dashed boxes.

GTA5 → Cityscapes: Table 1 shows that our proposed CCDA method performs much better on average than all state-of-the-art methods on the GTA5 → Cityscapes task. This advantage is derived from improvements over a wide range of classes. We achieved the best or second-best performances for nine classes, while for other classes, we were still able to reach results that are comparable to those of other methods. We also present visual comparisons of the proposed method with two other methods on this task in Fig. 3. This shows that, compared with the other methods, our method can not only provide

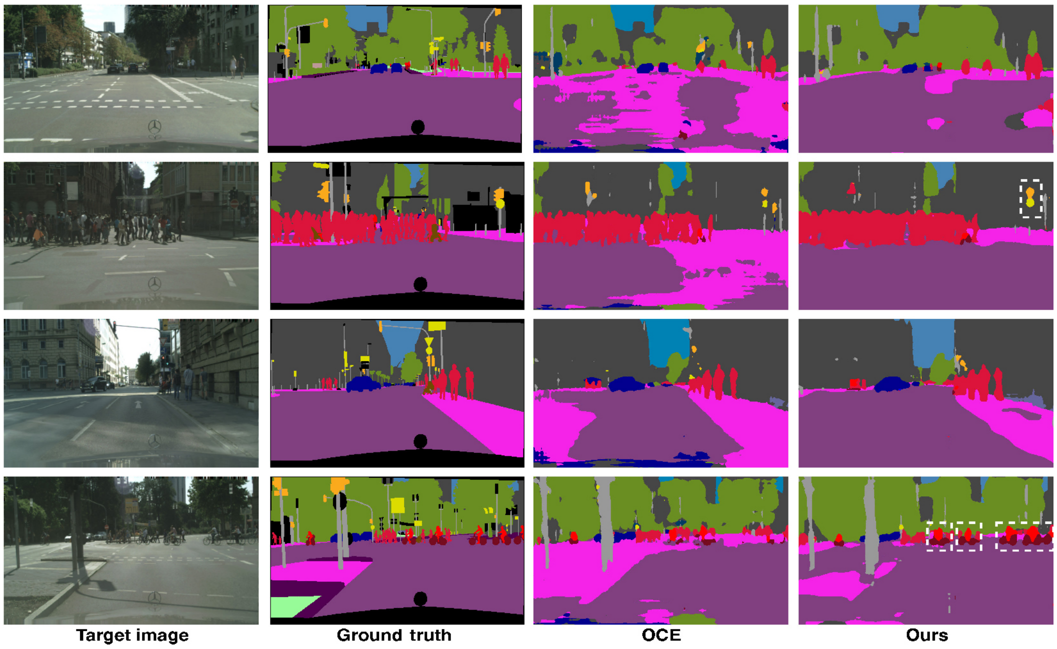


Fig. 4 Qualitative results of semantic segmentation on the SYNTHIA → Cityscapes task. For each target image, we show the corresponding ground-truth map and the results of OCE and our proposed CCDA method. We highlight some improved predictions with white dashed boxes.

cleaner predictions of more frequent classes, such as sidewalks and roads, but also improve the detection of less frequent classes, such as signs and lights. This demonstrates the effectiveness of the proposed CCDA method with a class-conditional multi-scale discriminator and class-conditional segmentation loss.

SYNTHIA→Cityscapes: Table 2 shows that our proposed CCDA method outperformed all methods in terms of mIoU over 16 classes on the SYNTHIA → Cityscapes transfer task. However, it achieved a slightly lower performance than SSF-DAN and OCE in terms of mIoU over 13 classes. Compared with the GTA5 dataset, the SYNTHIA dataset has fewer training images and may be more divergent from the Cityscapes dataset. Therefore, the SYNTHIA → Cityscapes transfer task was more difficult than the GTA5 → Cityscapes transfer task. To achieve explicit class-wise adaptation, the SSF-DAN method first separates the features into 19 parts for 19 classes and then applies 19 sub-discriminators, each of which can separately align features from each class. It is possible that the independent training of separate discriminators for each class performed by SSF-DAN makes it slightly better than our method (with a 0.2% improvement in mIoU* over 13 classes) for this more difficult task with a larger shift. However, this strategy requires the training of 19 sub-discriminators for 19 classes in the Cityscapes dataset. Because the model size of each sub-discriminator is normally approximately 9 MB, the overall model size of the 19 sub-discriminators in SSF-DAN could be approximately 170 MB. By contrast, our proposed class-conditional discriminator with two branches can employ a single discriminator system for all classes and occupy a much more economical 15 MB, which facilitates training and expansion to more categories. Therefore, the proposed CCDA method can be more efficient during training. It may also be more flexible than the sub-discriminator-based system of SSF-DAN for datasets with more categories because we need only change the number of output channels in the coarse-scale branch of the proposed class-conditional discriminator.

OCE also slightly outperformed our method in terms of mIoU* over 13 classes on the SYNTHIA → Cityscapes transfer task. It achieved feature alignment using contrastive learning, which aims to maximize the distance between different classes in the feature space. This strategy could achieve promising results for this more challenging transfer task by better distinguishing classes with lower frequency or smaller objects, such as motorbikes and riders. However, it outperformed our method by only 0.5% in terms of mIoU* over 13 classes on the SYNTHIA → Cityscapes task, while for mIoU over 16 classes on the SYNTHIA → Cityscapes task, OCE was 0.1%

worse than our proposed method. This suggests that consistently superior results for mIoU* and mIoU for this transfer task could be achieved by incorporating contrastive learning into our class-conditional adaptation system, which could be a promising topic for future work.

Compared to OCE, our method performed better on 9 out of 16 classes for the SYNTHIA → Cityscapes task. For the GTA5 → Cityscapes transfer task, OCE was 5.8% worse than our method in mIoU over 19 classes and achieved worse results than our method on almost all classes. We believe that this large improvement is primarily because our method applies the class-conditional approach for both feature extraction and feature alignment, which generally improves the overall performance. In addition, our proposed CCDA method may have a better generalization ability for different transfer tasks and classes compared to OCE. On the SYNTHIA → Cityscapes transfer task, the proposed method achieved the best or second-best performance for two classes compared with other state-of-the-art methods. Our method also achieved results comparable to those of the other methods over a large range of classes for this transfer task, which led to a better overall performance. We also present a visual comparison of our method with OCE on this task in Fig. 4. This indicates that our method provides cleaner predictions on more frequent classes, such as sidewalks and roads, and improved detection on less frequent classes, such as bicycles.

In general, we observe that compared with other methods that boost the performance for a few classes while sacrificing the performance for other classes, our class-conditional method often boosts the performance of almost all classes. Thus, although our method may not achieve the best performance among all classes, it ultimately achieves the capacity to generate higher mean IoU performance overall.

Complementarity with image translation: Meanwhile, we noticed that the image translation technique, which lowers the image style differences between domains, is complementary to methods with adaptation on representation. This has been applied to several recent state-of-the-art domain adaptation methods [27, 48] to generate translated source domain images in the target style to further alleviate the shift between the two domains at the input image level. Although our method focuses primarily on domain adaptation on representation, we also conducted experiments on the GTA5 → Cityscape task to demonstrate the complementarity of the proposed CCDA method with an image translation technique. For simplicity, we adopted the translated GTA5 images generated by Ref. [27] together with the original GTA5 images as the source domain input images to train our model. Meanwhile, we applied two rounds of self-supervised learning following Refs. [27, 28, 49] to enhance the performance. All other settings in this experiment were the same as those described in Section 4.1.

We compared our method with six state-of-the-art methods using image translation techniques: BDL [27], SEDA [50], LTIR [28], ITRA [48], CRA [31], and DPL [49]. Here, LTIR and CRA combine their proposed structures with existing image translation techniques (e.g., BDL), while others train their own image translation modules. The quantitative results of our method ("Ours+IT") with these state-of-the-art methods using VGG16 as the backbone are listed in Table 3. Note that we used image translation in a very simple manner. We did not use two types of style-transferred source domain images like LTIR does, and we did not use a teacher-student network that requires larger memory and computational costs during training, like SEDA does. Our proposed CCDA with image translation still outperformed most state-of-the-art methods, which demonstrates the complementarity of the proposed CCDA method with image translation.

Our performance was only worse than that of DPL because DPL applies a dual-path learning strategy by training two image translation networks with two corresponding segmentation networks. By training the image translation and segmentation networks instead of directly using translated images from an existing method as ours does, the translated images generated by DPL are more suitable for its segmentation networks. Meanwhile, DPL requires two image translation networks to not only generate translated source domain images with target style, as BDL and ITRA do, but also produce translated target domain images with the source style. Therefore, using the translated images from the two domains definitely helps to better alleviate the shift between the two domains, which

Table 3 Adaptation from GTA5 to Cityscapes with extra image translation (IT) technique. We present the per-class and mean IoU, and we highlight the best result in each column in bold font

	GTA5 → Cityscapes																			
	road	side.	buil.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	mbike.	bike	mIoU
BDL	89.2	40.9	81.2	29.1	19.2	14.2	29.0	19.6	83.7	35.9	80.7	54.7	23.3	82.7	25.8	28.0	2.3	25.7	19.9	41.3
SEDA	90.2	51.5	81.1	15.0	10.7	**37.5**	35.2	**28.9**	84.1	32.7	75.9	**62.7**	19.9	82.6	22.9	28.3	0.0	23.0	25.4	42.5
LTIR	92.5	54.5	**83.9**	34.5	25.5	31.0	30.4	18.0	84.1	**39.6**	**83.9**	53.6	19.3	81.7	21.1	13.6	**17.7**	12.3	6.5	42.3
ITRA	89.2	43.4	80.0	30.1	19.4	27.8	27.1	13.3	80.5	35.8	71.2	50.6	20.7	80.2	26.8	33.4	0.0	17.7	11.9	39.9
CRA	89.1	42.0	81.2	28.9	23.1	13.9	29.3	17.0	83.6	36.6	81.7	56.2	25.5	81.9	26.0	32.0	0.2	26.8	19.7	41.8
Ours+IT	**92.9**	**56.6**	82.5	31.0	**27.1**	31.0	36.2	27.0	80.8	32.1	73.9	57.3	21.5	82.1	18.5	21.8	10.4	22.6	12.1	43.0
DPL	89.2	44.0	83.5	**35.0**	24.7	27.8	**38.3**	25.3	**84.2**	39.5	81.6	54.7	**25.8**	**83.3**	**29.3**	**49.0**	5.2	**30.2**	**32.6**	**46.5**

largely improves its performance compared with other methods. Additionally, the dual-path learning framework allows complementary information to interact between the two paths. However, this requires the training of two different image translation networks and two different segmentation networks for two paths, which significantly increases the memory and computational costs during training. During testing, DPL also requires the generation of translated target domain images with the source style in advance. The translated target domain images are then used as inputs to obtain the segmentation predictions. This helps to further improve the performance; however, it also affects the efficiency of DPL during testing.

4.3 Ablation studies

Ablation study on different components: To better understand the impact of each component of our adaptation model, we conducted an ablation study by selectively deactivating each component and measuring the effect on the performance of the GTA5 → Cityscapes transfer task. Specifically, we defined four nested subset models:

(1) B: Using the basic domain adaptation architecture in Section 3.1 with segmentation loss (Eq. (1)) and a fine-scale basic discriminator for adaptation (Eqs. (2) and (3)). This means that we set $\alpha = 1$ and $\beta = 1$ in the blend losses.

(2) $\mathrm{B} + \mathrm{S_c}$: Adding class-conditional loss for segmentation in Section 3.3, where $\alpha = 0.5$ in Eq. (17).

(3) $\mathrm{B} + \mathrm{S_c} + \mathrm{D_c}$: Further adding the class-conditional discriminator in Section 3.2, including the coarse-scale branch loss (Eqs. (4) and (5)) and the class-conditional fine-scale branch loss by setting $\beta = 0.4$ in Eqs. (12) and (13).

(4) Ours: Our final model with the extra entropy minimization loss in Eq. (18).

The results are presented in Table 4 showing that our overall CCDA system resulted in a performance gain of 3.1% over the basic domain adaptation architecture in mIoU. The class-conditional segmentation loss alone was responsible for a 1.5% improvement, the class-conditional discriminator produced an additional 1.3% improvement, and the entropy minimization loss provides a slight 0.3% improvement. This verifies the importance of both the class-conditional segmentation and discriminator components of our CCDA approach. Examples of qualitative segmentation are shown in Fig. 5.

To analyze the impact of each component on the

Table 4 Ablation study of our CCDA method on GTA5 → Cityscape task. The numbers above all classes are the indexes of frequency in descending order for Cityscapes. We highlight the best result in each column in bold font

	GTA5 → Cityscapes																			
Method	1	5	2	11	10	7	17	12	3	9	6	8	18	4	15	14	16	19	13	
	road	side.	buil.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	mbike.	bike	mIoU
B	86.7	37.8	79.6	**30.3**	21.7	29.0	26.8	17.1	79.2	20.8	72.6	50.0	5.8	79.6	**23.2**	18.9	0.0	19.6	2.5	36.9
$\mathrm{B} + \mathrm{S_c}$	85.1	37.6	79.8	25.6	21.1	29.2	**32.0**	**23.2**	78.7	21.3	71.7	54.4	14.6	79.6	19.9	17.2	**7.1**	**23.7**	**8.8**	38.4
$\mathrm{B} + \mathrm{S_c} + \mathrm{D_c}$	90.6	**45.6**	80.8	26.6	**23.5**	**30.2**	31.9	21.8	79.1	24.4	74.3	**55.0**	**16.1**	81.1	20.4	**20.3**	5.9	20.7	5.7	39.7
Ours	**91.1**	45.1	**81.1**	29.8	23.2	30.1	31.2	19.7	**81.3**	**29.4**	**74.5**	54.0	15.6	**81.4**	21.3	**20.3**	3.8	21.7	6.0	**40.0**

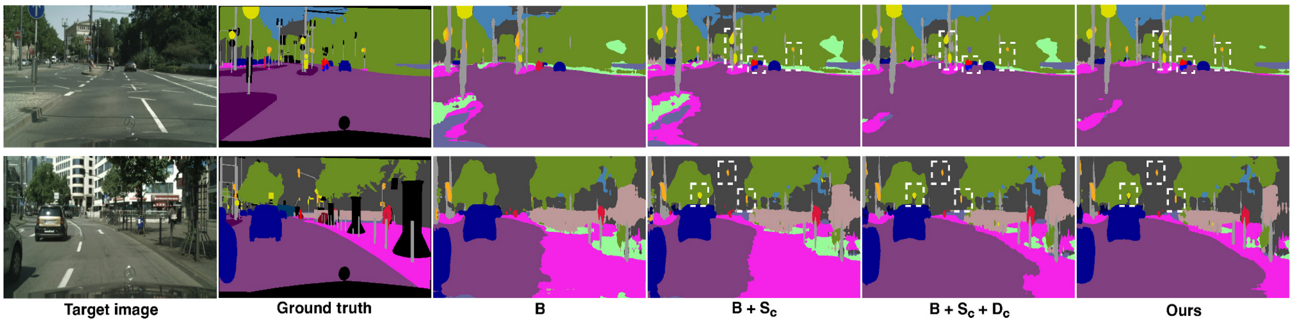


Fig. 5 Qualitative results of ablation study on the GTA5 → Cityscapes task. For each target image, we show the corresponding ground-truth map and the results of each subset model in the ablation study. We highlight some improved predictions with white dashed boxes.

different classes more effectively, we also present the frequency of each class in the Cityscapes dataset in descending order in Table 4. Measuring the segmentation prediction in a class-wise manner in "$\mathrm{B}+\mathrm{S}_\mathrm{c}$" tends to improve the performance on some less frequent classes, such as lights and bikes. This is reasonable because using class-conditional segmentation loss may increase the weight of the loss on classes with lower frequencies or objects with smaller sizes. This improves the overall performance and prevents the model from neglecting these classes during adaptation. However, it may also sacrifice performance in more frequent classes, such as roads and skies. By further adding our class-conditional discriminator as in "$\mathrm{B}+\mathrm{S}_\mathrm{c}+\mathrm{D}_\mathrm{c}$", it promotes a class-wise adaptation with the design of our coarse- and fine-scale branches. This achieves improvements on both more frequent classes, such as roads and sidewalks, and less frequent classes, such as riders and buses. "Ours" with the extra entropy minimization loss further helps to balance the performance on different classes and slightly improve the overall performance. Compared to the basic model "B", our overall CCDA system ("Ours") achieved improvements on almost all classes.

Figure 5 also proves the effectiveness of each component of our CCDA system. Compared to "B", "$\mathrm{B}+\mathrm{S}_\mathrm{c}$" performed better on some smaller objects and less frequent classes, such as signs and lights. However, for some larger objects and more frequent classes, such as sidewalks and roads, the performance of "$\mathrm{B}+\mathrm{S}_\mathrm{c}$" may not be improved or even become worse. "$\mathrm{B}+\mathrm{S}_\mathrm{c}+\mathrm{D}_\mathrm{c}$" can further improve the performance on some less frequent classes, such as terrain, while achieving improvements on some larger objects and more frequent classes compared to "$\mathrm{B}+\mathrm{S}_\mathrm{c}$". "Ours" can further slightly improve the results by balancing the performance among different classes. Therefore, our overall CCDA system enhances the performance through general improvements on various classes.

Ablation study on thresholds: We apply two thresholds in our method: th_w and th_a. To avoid neglecting any classes that occur in each target domain patch for achieving the coarse-scale class labels used in the class-conditional coarse-scale discriminator branch, we set the threshold $th_w = 0.4$. To avoid ignoring any ambiguous pixels in the target domain images for the class-conditional fine-scale discriminator branch, we set the threshold $th_\mathrm{a} = 0.95$. To explore the sensitivity of the two thresholds, we conducted experiments on the GTA5 → Cityscapes task by setting different th_w and th_a. The results are presented in Tables 5 and 6, which show that by setting these thresholds within reasonable ranges, the overall performance of the proposed method is not significantly affected.

4.4 Limitations and future work

As illustrated by the previous experimental results (Fig. 3 and Fig. 4), the proposed CCDA method performs well in most situations. However, it still has some limitations; for example, it may fail to detect some objects with very small size, or it may be

Table 5 Sensitivity analysis of threshold th_w

GTA5 → Cityscapes task					
th_w	0.2	0.3	**0.4**	0.5	0.6
mIoU	39.7	39.8	**40.0**	39.6	39.4

Table 6 Sensitivity analysis of threshold th_a

GTA5 → Cityscapes task					
th_a	0.80	0.85	0.90	**0.95**	0.98
mIoU	39.6	39.7	39.9	**40.0**	39.6

unable to predict clear edges for large objects under complicated scenarios. Therefore, in future work, we will investigate how to improve the performance by training our model with the help of multiple related tasks, such as depth estimation, object detection, and boundary detection. These related tasks can be beneficial to the segmentation model by providing additional information. For example, an object detection task [51, 52] may assist the network in maintaining semantic information for smaller objects, while boundary detection [53] can be helpful for learning a clearer contour for each object. Moreover, video segmentation [54, 55] is a challenging task, and we plan to investigate how our proposed class-conditional method can be extended to video semantic segmentation in an unsupervised manner, as well as more applications with various datasets.

5 Conclusions

We developed a novel approach to solve an important problem for domain adaptation for semantic segmentation in that the different abilities for representation extraction and alignment for different classes may affect the adaptation performance. The solution hinges on the introduction of class-conditioning at multiple points in the model, including a class-conditional segmentation loss and class-conditional multi-scale discriminator, which measure the segmentation prediction and adaptation in a class-wise manner. The experimental results of the ablation study demonstrate that our overall CCDA method improves performance for almost all classes and boosts overall performance. Extensive experimental results demonstrate the effectiveness of our method by reaching comparable results, and in some cases, outperforming state-of-the-art methods.

Acknowledgements

We would like to thank the York University Vision: Science to Applications (VISTA) program and Intelligent Systems for Sustainable Urban Mobility (ISSUM) project, funded by the Ontario Research Fund-Research Excellence program for their supports.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Gong, L. X.; Zhang, Y. Q.; Zhang, Y. K.; Yang, Y.; Xu, W. W. Erroneous pixel prediction for semantic image segmentation. *Computational Visual Media* Vol. 8, No. 1, 165–175, 2022.

[2] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440, 2015.

[3] Zhao, H. S.; Shi, J. P.; Qi, X. J.; Wang, X. G.; Jia, J. Y. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6230–6239, 2017.

[4] Wang, W. H.; Xie, E. Z.; Li, X.; Fan, D. P.; Song, K. T.; Liang, D.; Lu, T.; Luo, P.; Shao, L. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media* Vol. 8, No. 3, 415–424, 2022.

[5] Yao, T.; Pan, Y. W.; Ngo, C. W.; Li, H. Q.; Mei, T. Semi-supervised Domain Adaptation with Subspace Learning for visual recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2142–2150, 2015.

[6] Tsai, Y. H.; Hung, W. C.; Schulter, S.; Sohn, K.; Yang, M. H.; Chandraker, M. Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7472–7481, 2018.

[7] Tsai, Y. H.; Sohn, K.; Schulter, S.; Chandraker, M. Domain adaptation for structured output via discriminative patch representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1456–1465, 2019.

[8] Chen, Y. H.; Chen, W. Y.; Chen, Y. T.; Tsai, B. C.; Wang, Y. C F.; Sun, M. No more discrimination: Cross city adaptation of road scene segmenters. In: Proceedings of the IEEE International Conference on Computer Vision, 2011–2020, 2017.

[9] Luo, Y. W.; Liu, P.; Guan, T.; Yu, J. Q.; Yang, Y. Significance-aware information bottleneck for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6777–6786, 2019.

[10] Sun, B. C.; Feng, J. S.; Saenko, K. Return of frustratingly easy domain adaptation. In: Proceedings of the 30th AI Conference on Artificial Intelligence, 2058–2065, 2016.

[11] Geng, B.; Tao, D. C.; Xu, C. DAML: Domain

adaptation metric learning. *IEEE Transactions on Image Processing* Vol. 20, No. 10, 2980–2989, 2011.

[12] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2672–2680, 2014.

[13] Zhou, W.; Wang, Y. K.; Chu, J. J.; Yang, J. H.; Bai, X.; Xu, Y. C. Affinity space adaptation for semantic segmentation across domains. *IEEE Transactions on Image Processing* Vol. 30, 2549–2561, 2021.

[14] Vu, T. H.; Jain, H.; Bucher, M.; Cord, M.; Perez, P. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2512–2521, 2019.

[15] Shan, Y. H.; Chew, C. M.; Lu, W. F. Semantic-aware short path adversarial training for cross-domain semantic segmentation. *Neurocomputing* Vol. 380, 125–132, 2020.

[16] Rozantsev, A.; Salzmann, M.; Fua, P. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 41, No. 4, 801–814, 2019.

[17] Sun, B.; Saenko, K. Deep CORAL: Correlation alignment for deep domain adaptation. In: *Computer Vision – ECCV 2016 Workshops. Lecture Notes in Computer Science, Vol. 9915.* Hua, G.; Jégou, H. Eds. Springer Cham, 443–450, 2016

[18] Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2962–2971, 2017.

[19] Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, Vol. 37, 1180–1189, 2015.

[20] Hoffman, J.; Wang, D.; Yu, F.; Darrell, T. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint* arXiv:1612.02649, 2016.

[21] Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning, Vol. 70, 214–223, 2017.

[22] Choi, Y.; Choi, M.; Kim, M.; Ha, J. W.; Kim, S.; Choo, J. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8789–8797, 2018.

[23] Harms, J.; Lei, Y.; Wang, T.; Zhang, R.; Zhou, J.; Tang, X.; Curran, W. J.; Liu, T.; Yang, X. Paired cycle-GAN-based image correction for quantitative cone-beam computed tomography. *Medical Physics* Vol. 46, No. 9, 3998–4009, 2019.

[24] Isola, P.; Zhu, J. Y.; Zhou, T. H.; Efros, A. A. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5967–5976, 2017.

[25] Zou, Y.; Yu, Z. D.; Vijaya Kumar, B. V. K.; Wang, J. S. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11207.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 297–313, 2018.

[26] Zhang, Y.; David, P.; Foroosh, H.; Gong, B. Q. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 42, No. 8, 1823–1841, 2020.

[27] Li, Y. S.; Yuan, L.; Vasconcelos, N. Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6929–6938, 2019.

[28] Kim, M.; Byun, H. Learning texture invariant representation for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12972–12981, 2020.

[29] Liu, Y. A.; Zhang, W.; Wang, J. Source-free domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1215–1224, 2021.

[30] Klingner, M.; Termohlen, J. A.; Ritterbach, J.; Fingscheidt, T. Unsupervised BatchNorm adaptation (UBNA): A domain adaptation method for semantic segmentation without using source domain representations. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, 210–220, 2022.

[31] Zhang, X. H.; Chen, Y.; Shen, Z. Y.; Shen, Y. M.; Zhang, H. F.; Zhang, Y. D. Confidence-and-refinement adaptation model for cross-domain semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems* Vol. 23, No. 7, 9529–9542, 2022.

[32] Luo, Y. W.; Liu, P.; Zheng, L.; Guan, T.; Yu, J.

Q.; Yang, Y. Category-level adversarial adaptation for semantic segmentation using purified features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 44, No. 8, 3940–3956, 2022.

[33] Du, L.; Tan, J. G.; Yang, H. Y.; Feng, J. F.; Xue, X. Y.; Zheng, Q. B.; Ye, X. Q.; Zhang, X. L. SSF-DAN: Separated semantic feature based domain adaptation network for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 982–991, 2019.

[34] Yang, J. H.; Xu, R. J.; Li, R. Y.; Qi, X. J.; Shen, X. Y.; Li, G. B.; Lin, L. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 34, No. 7, 12613–12620, 2020.

[35] Milletari, F.; Navab, N.; Ahmadi, S. A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the 4th International Conference on 3D Vision, 565–571, 2016.

[36] Nie, D.; Gao, Y. Z.; Wang, L.; Shen, D. G. ASDNet: Attention based semi-supervised deep networks for medical image segmentation. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. Lecture Notes in Computer Science, Vol. 11073.* Frangi, A.; Schnabel, J.; Davatzikos, C.; Alberola-López, C.; Fichtinger, G. Eds. Springer Cham, 370–378, 2018.

[37] Wong, K. C. L.; Moradi, M.; Tang, H.; Syeda-Mahmood, T. 3D segmentation with exponential logarithmic loss for highly unbalanced object sizes. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. Lecture Notes in Computer Science, Vol. 11072.* Frangi, A.; Schnabel, J.; Davatzikos, C.; Alberola-López, C.; Fichtinger, G. Eds. Springer Cham, 612–619, 2018.

[38] Toldo, M.; Michieli, U.; Zanuttigh, P. Unsupervised domain adaptation in semantic segmentation via orthogonal and clustered embeddings. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 1357–1367, 2021.

[39] Richter, S. R.; Vineet, V.; Roth, S.; Koltun, V. Playing for data: Ground truth from computer games. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9906.* Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 102–118, 2016.

[40] Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A. M. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3234–3243, 2016.

[41] Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3213–3223, 2016.

[42] Chen, L. C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 40, No. 4, 834–848, 2018.

[43] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the 3rd International Conference on Learning Representations, 2015.

[44] Maas, A. L.; Hannun, A. Y.; Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

[45] Bottou, L. Large-scale machine learning with stochastic gradient descent. In: Proceedings of the COMPSTAT' 2010, 177–186, 2010.

[46] Kingma, D. P.; Ba, J. L. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, 2015.

[47] Zhang, X. H.; Zhang, H. F.; Lu, J. F.; Shao, L.; Yang, J. Y. Target-targeted domain adaptation for unsupervised semantic segmentation. In: Proceedings of the IEEE International Conference on Robotics and Automation, 13560–13566, 2021.

[48] Kang, J. X.; Zang, B.; Cao, W. P. Domain adaptive semantic segmentation via image translation and representation alignment. In: Proceedings of the IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking, 509–516, 2021.

[49] Cheng, Y. T.; Wei, F. Y.; Bao, J. M.; Chen, D.; Wen, F.; Zhang, W. Q. Dual path learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9062–9071, 2021.

[50] Choi, J.; Kim, T.; Kim, C. Self-ensembling with GAN-based data augmentation for domain adaptation in semantic segmentation. In: Proceedings of the

IEEE/CVF International Conference on Computer Vision, 6829–6839, 2010.

[51] Liang, T. T.; Chu, X. J.; Liu, Y. D.; Wang, Y. T.; Tang, Z.; Chu, W.; Chen, J. D.; Ling, H. B. CBNet: A composite backbone network architecture for object detection. *IEEE Transactions on Image Processing* Vol. 31, 6893–6906, 2022.

[52] Lan, Y. Q.; Duan, Y.; Liu, C. Y.; Zhu, C. Y.; Xiong, Y. S.; Huang, H.; Xu, K. ARM3D: Attention-based relation module for indoor 3D object detection. *Computational Visual Media* Vol. 8, No. 3, 395–414, 2022.

[53] Liu, Y.; Xie, Z. W.; Liu, H. An adaptive and robust edge detection method based on edge proportion statistics. *IEEE Transactions on Image Processing* Vol. 29, 5206–5215, 2020.

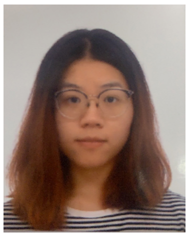
[54] Ji, G. P.; Fan, D. P.; Fu, K. R.; Wu, Z.; Shen, J. B.; Shao, L. Full-duplex strategy for video object segmentation. *Computational Visual Media* Vol. 9, No. 1, 155–175, 2023.

[55] You, M. Y.; Luo, C. X.; Zhou, H. J.; Zhu, S. Q. Dynamic dense CRF inference for video segmentation and semantic SLAM. *Pattern Recognition* Vol. 133, 109023, 2023.

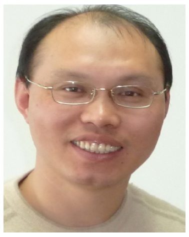
Yue Wang is a Ph.D. student in Signal and Information Processing, Dalian University of Technology. Her research interest is in saliency detection and unsupervised learning.

Yuke Li received his Ph.D. degree in communication and information system, Wuhan University. His research interests include computer vision and deep learning.

James H. Elder is presently a professor in the Department of Electrical Engineering and Computer Science and the Department of Psychology, York University. His research interests include shape perception, single-view 3D reconstruction.



Runmin Wu is currently studying in computer science, the University of Hong Kong. Her research interest is in computer vision.



Huchuan Lu is a professor in the Department of Electronic Information and Electrical Engineering, Dalian University of Technology. His recent research interests focus on computer vision, artificial intelligence, pattern recognition, and machine learning.

Other papers from this open access journal are available free of charge from http://www.springer.com/journal/41095. To submit a manuscript, please go to https://www.editorialmanager.com/cvmj.