# Domain-invariant Prototypes for Semantic Segmentation

Zhengeng Yang, Hongshan Yu, Wei Sun, Li-Cheng, Ajmal Mian

***Abstract*—Deep Learning has greatly advanced the performance of semantic segmentation, however, its success relies on the availability of large amounts of annotated data for training. Hence, many efforts have been devoted to domain adaptive semantic segmentation that focuses on transferring semantic knowledge from a labeled source domain to an unlabeled target domain. Existing self-training methods typically require multiple rounds of training, while another popular framework based on adversarial training is known to be sensitive to hyper-parameters. In this paper, we present an easy-to-train framework that learns domain-invariant prototypes for domain adaptive semantic segmentation. In particular, we show that domain adaptation shares a common character with few-shot learning in that both aim to recognize some types of unseen data with knowledge learned from large amounts of seen data. Thus, we propose a unified framework for domain adaptation and few-shot learning. The core idea is to use the class prototypes extracted from few-shot annotated target images to classify pixels of both source images and target images. Our method involves only one-stage training and does not need to be trained on large-scale un-annotated target images. Moreover, our method can be extended to variants of both domain adaptation and few-shot learning. Experiments on adapting GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes show that our method achieves competitive performance to state-of-the-art.**

***Index Terms*—Semantic Segmentation, Domain Adaptation, Few-shot Learning, Prototype Learning, Metric Learning.**

## I. Introduction

SEMANTIC segmentation is the task of assigning each pixel a semantic label, such as car, bicycle, tree, among others. Recent deep learning methods (e.g. [1], [2]) have significantly advanced the state-of-the-art performance in semantic segmentation. However, their performance heavily relies on the availability of large-scale labelled training datasets. Unfortunately, for semantic segmentation, this entails pixel-level annotations that is often an extremely time consuming and tedious process. For example, Saleh et al. [3] mention that it takes on average 90 minutes to manually annotate a single $1024 \times 2048$ image for semantic segmentation.

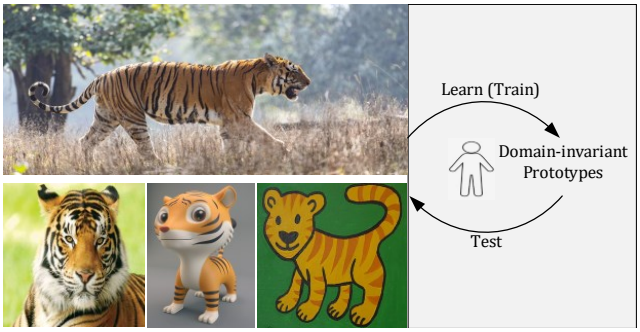


Fig. 1. The main hypothesis that motivates our method. The impressive domain adaptation ability of humans (we can still recognize the second row images as tiger) originates from the fact that we learn domain-invariant class prototypes, and then perform object classification through a "similarity comparison" process with respect to the prototypes.

Numerous research attempts [4], [5] have been made to avoid the manual annotation by leveraging computer graphics techniques to automatically generate synthetic images along with their corresponding pixel-wise labels. However, models directly trained on synthetic images often do not perform well on real data, which is largely attributed to the issue of domain shift. Therefore, domain adaptive semantic segmentation that focuses on transferring semantic knowledge from a labeled source domain to an unlabeled target domain has attracted significant research attention over the past decade [6]–[10]. Although current self-training based methods and adversarial training based methods methods have achieved great success in domain adaptive semantic segmentation, these two popular approaches usually involve a cumbersome training process. In particular, both need to pre-train an initial segmentation model on the source domain. Additionally, self-training based methods generally perform self-training on the entire target domain for multiple rounds, while GAN-based methods are known to be difficult to train due to their sensitivity to hyper-parameters.

Besides adaptation from virtual domain, another direction of alleviating the reliance on large-scale manual annotations is to use the few-shot learning, which aims to learn to recognize some query classes with only a few (e.g., 1,5) annotated support samples of them. Towards this goal, the few-shot learning typically learns the model over a set of base classes that has already been associated with a large number of annotated samples, while the challenging few-shot scenario is only applied to unseen classes. Since the few-shot learning

This work was supported in part by the National Natural Science Foundation of China under Grant 62103137, U2013203, 61973106, U1913202, Professor Ajmal Mian is the recipient of an Australian Research Council Future Fellowship Award (project number FT210100268) funded by the Australian Government.

Zhengeng Yang, Hongshan Yu, and Wei Sun are with the National Engineering Laboratory for Robot Visual Perception and Control Technology, College of Electrical and Information Engineering, Hunan University, Lushan South Rd., Yuelu Dist., 410082, Changsha, China. Zhengeng Yang and Hongshan Yu contributed equally to this work. H. Yu is the corresponding author ( e-mail: yuhongshancn@hotmail.com).

Li Cheng is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada.

Ajmal Mian is with the Department of Computer Science, The University of Western Australia, Perth, WA 6009, Australia

provides a solution to extend the learned model to unseen classes, it has received great attention in recent several years, especially the prototype-based method [11]–[13]. The core idea of prototype based few-shot learning is to treat the average features of the few-shot support samples as the prototypes of unseen classes. Then, the classification of the query samples can be performed by measuring the feature distances to these prototypes and choosing the nearest class prototype as the result.

Inspired by the success of prototype-based class representation in few-shot learning, we hypothesize that the impressive domain adaptation ability of humans (Fig. 1) originates from the fact that we learn domain-invariant class prototypes, and then perform object classification through a "similarity comparison" process with respect to the prototypes. Based on this, we propose to formulate the domain adaptation as a domain-invariant prototype learning problem. Toward this goal, we first extract class prototypes from a few target images with annotations, and then use them to segment the source images. The prototype learning thus can be supervised by the large amount of source annotations. After that, the segmentation on arbitrary target images is implemented by computing pixel-wise similarity with these learned class prototypes. As shown in Fig. 2, compared with using self-training and adversarial training, our few-shot domain adaptation involves only one stage training and need not to be pre-trained on the source domain. Moreover, our method also need not to access large scale un-annotated target images.

Since our few-shot domain adaption is developed from current popular prototype-based few-shot segmentation framework, it is noteworthy that existing few-shot segmentation methods usually consider a binary segmentation task on a specific class during training. The reasons for this consideration will be detailed in Section III-B. Although this training strategy works well for low-resolution images containing a few salient objects (e.g., images in PASCAL dataset), it is not suitable for high-resolution images containing arbitrary number of classes for two reasons. Firstly, there are many object classes (e.g., traffic light) that occupy only a small fraction of the entire image. Binary segmentation on these classes means all other objects will be treated as background, which is harmful for the feature representation learning. Secondly, it is difficult to ensure those small object classes are still contained in the training image after the common random image cropping data augmentation is applied to images of high-resolution. In other words, the classes contained in the training image cropped from high-resolution images are uncertain. Thus, we cannot perform a binary segmentation on a specific class. Based on the observation that the prototypes used for class recognition are extracted from support images, we present a support image depended training strategy in which a class will be recognized if and only if it appears in current support images. Thus, the classes to be recognized in each training step are essentially adapted to the randomly sampled training images, which makes our few-shot learning method can be generalized to most scenes without constraints on image resolution and saliency of objects.

To summarize, our contributions are three folds as follows.

First, we present a novel domain adaptation framework that formulates domain adaptation as a domain-invariant prototype learning problem. Compared with current methods based on self-training and generative adversarial networks (GAN) [14], our method is much easier to implement since it did not require to either perform multiple rounds training or optimize multiple adversarial losses.

Second, we extend the current few-shot segmentation problem to a more generalized form. Instead of segmenting only "things" (objects) from the background like most few-shot segmentation methods do, the proposed method considers segmenting "things" and "stuffs" simultaneously. Moreover, we present a support image adaptive training strategy for few-shot learning without constraints on image resolution and saliency of objects. To the best of our knowledge, we are the first to consider few-shot segmentation task for high resolution images.

Third, we developed a unified framework for domain adaptation and few-shot learning, where unified means our method can be applied to both of these two tasks.

## II. Related Work

### A. Domain Adaptive Semantic Segmentation

To alleviate the requirement for human-labeled images, a recent trend for semantic segmentation is to use synthetic images where dense annotations are automatically available [8], [10], [15], [16]. However, models trained on synthetic data when applied directly to real data do not perform well due to the distribution shift between the two domains. This gives rise to the challenging problem of domain adaptive semantic segmentation. In general, existing domain adaptive semantic segmentation methods can be roughly divided into two categories: self-training based methods and adversarial training based methods.

**Self-training based methods:** The core idea of this line of methods is to generate pseudo labels in the target domain, where the pseudo labels are usually obtained by taking the predicted labels with high confidence. Zou et al. [17] were among the first to apply self-training to domain adaptive semantic segmentation. They organized the self-training into a two-stage process (which we name as select-and-train) which is repeated for multiple rounds. Many following works focus on improving the quality of pseudo labels [6]–[9]. For examples, Li et al. [6] proposed to use only synthetic images that share similar distribution with the real images for domain adaptation, Zheng et al. [7] developed an uncertainty-based adaptive threshold for pseudo label selection. Since the confidence based pseudo label selection can easily ignore hard samples, another interest of self-training based methods is to prevent the training process being dominated by easy samples [18], [19]. Typical ways for solving this problem include designing robust losses [18] and employing multiple classifiers [7].

**Adversarial training based methods:** These methods employ adversarial training to align the source domain and target domain either at feature-level, image-level, or both. Adversarial feature alignment generally follows the GAN [14]

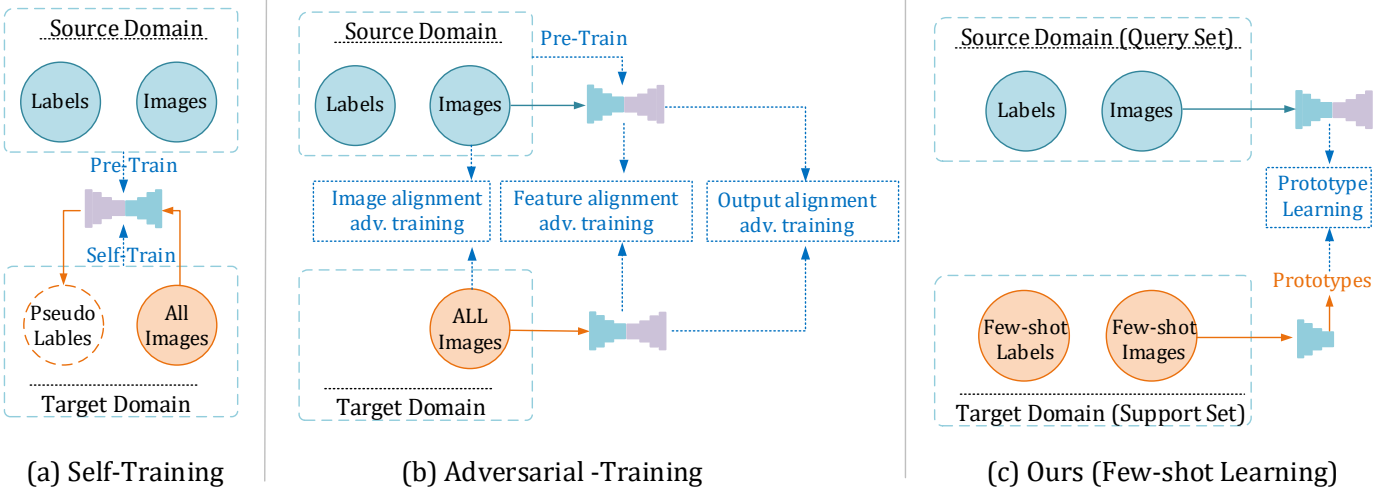


Fig. 2. The proposed framework compared to two popular frameworks. Self-training and adversarial-training methods generally involve cumbersome training processes. The former usually needs to perform self-training on the entire target dataset for multiple rounds, while the latter needs to optimize multiple adversarial losses which are sensitive to hyper parameters. In contrast, our proposed method learns domain-invariant prototypes with the support of few-shot target annotations. Our method involves only one-stage training and need not to access large-scale un-annotated target images.

framework. In particular, by treating the feature encoder as the generator, adversarial feature alignment [20]–[22] tries to fool a discriminator that distinguishes whether the feature is from the source domain or the target domain. Instead of aligning intermediate features, image-level alignment [23] directly mitigates the domain gap in the image space with image-to-image translation [24]. More recently, aligning jointly image space, intermediate features, and output space [25]–[27] is becoming popular. Although adversarial training provides an effective way to mitigate the domain gap problem, it suffers from several drawbacks such as being computationally expensive and difficult to train given its sensitivity to hyper-parameters.

### B. Few-shot Supervised Domain Adaptation

Most of the above methods consider an unsupervised domain adaptation problem, i.e., they assume that there are no labeled target images. Inspired by few-shot learning, we relax this setting and assume that there are a few labeled target images available and then propose a few-shot domain adaptation framework. Note that there are some few-shot unsupervised domain adaptation methods [28], [29] in which the few-shot scenario is applied over the source domain, i.e., there are only a few labeled source images available and the target domain is still unlabeled. This paper essentially considers a few-shot supervised domain adaption problem that has also been explored in many tasks. For example, Motiian et al. [30] developed a few-shot adversarial domain adaptation method for image classification. For the semantic segmentation task, Zhang et al. [31] proposed to use a few target annotations to enhance the semantic feature learning on target domain during domain adversarial learning. Thus, these related methods can be still categorized into adversarial training based methods mentioned above. In contrast, we propose to solve the few-shot domain adaptation by the commonly used prototype-based few-shot learning framework. To the best of our knowledge, we are the first to formulate the domain adaption as a few-shot learning problem.

### C. Few-shot Semantic Segmentation

Few-shot learning provides another way to reduce the reliance on human-labeled samples by recognizing novel classes with minimal support i.e., labeled samples. The first work [32] on few-shot segmentation proposed to adjust the final classifier with weights learned from support samples.

Recently, applying metric learning to classify images to the nearest class prototype extracted from support images has become popular in few-shot image classification [33]. Dong et al. [34] were among the first to apply such a prototype-based framework to few-shot segmentation. Following their work, PANet [35] introduced prototype alignment regularization by performing an additional query-to-support segmentation after the standard support-to-query segmentation. AMP [11] learned the prototypes from multi-scale features and historical support samples, and Li et al. [12] proposed to consider multiple prototypes for each class.

Instead of using prototype as the anchor feature for classification, CRNet [13] and PFENet [36] incorporated the prototypes into the query features and formulated the few-shot segmentation as a binary classification problem.

Our work is closely related to prototype-based few-shot segmentation methods such as PANet [35]. However, our work differs in three aspects. First, we focus on applying few-shot learning to solve the challenging domain adaptation task. Second, most existing few-shot segmentation methods evaluated their performance on PASCAL VOC 2012 dataset [37] and COCO dataset [38] and mainly focused on recognizing "things" [39] (objects such as car). In contrast, we consider a more general few-shot learning problem by further recognizing "stuffs" (e.g., sky, road). Finally, existing few-shot segmentation methods consider a binary segmentation task for training,

i.e., recognizing only one class for each step, which requires objects of that specific class to occupy a certain proportion of the training image. Such a requirement can be easily satisfied when images have low-resolution and contain only a few objects. However, compared with images of PASCAL and COCO, the urban images considered in this paper have significantly higher resolution and contain many more objects of different classes. For this, we developed a support image adaptive training strategy (details in Section IV-D) suited for few-shot segmentation on high-resolution images that contain arbitrary number of objects.

### D. Prototype Learning in Domain Adaptation

Since the proposal of prototypical network [33] for few-shot learning, incorporating prototype learning into domain adaptation has also received increasing attention in recent years. For example, Zhang et al [8] presented a prototype based denoise method to improve the quality of pseudo labels for the self-training framework. The motivation of this method can be summarized as: the pixel with noisy label is typically far from the class prototype corresponding to the same noisy label in the embedding space. Yue et al. [28] developed a prototypical self-supervised learning framework by performing a cross-domain instance-prototype matching task. Different from these methods, in which the prototypes are usually used to facilitate the feature learning, our method focus on learn domain-invariant prototypes that will be directly used for classification. From this perspective, the work most related to ours is the transferable prototypical network (TPN) proposed by Pan et al. [40]. However, TPN is an unsupervised domain adaptation framework and learns the prototypes through minimizing domain discrepancy such as cross-domain prototype distances. In contrast, our method considers a few-shot supervised domain adaptation problem and the prototype learning are directly supervised by the classification loss.

## III. Preliminaries

Since our work focuses on the domain adaptive semantic segmentation task and is developed from prototype-based few-shot semantic segmentation, we start by introducing commonly used formulations in these two tasks.

### A. Domain Adaptive Semantic Segmentation

Domain adaptive semantic segmentation typically involves a source domain $S$ and a target domain $T$. The source domain contains a large number of images $I_s = \{I_j\}_{j=1}^{n_s}$ with dense annotations $Y_s = \{y_j\}_{j=1}^{n_s}$, where $n_s$ is the total number of images in $S$. The goal of domain adaptive semantic segmentation is to apply the knowledge learned from the source domain to the target dataset $I_t = \{I_j\}_{j=1}^{n_t}$ such that the requirement for annotations of target images is minimized.

### B. Few-shot Segmentation

Few-shot semantic segmentation is related to the few-shot learning which aims to recognize a set of novel test classes $C_{test}$ by learning models from a set of base training classes

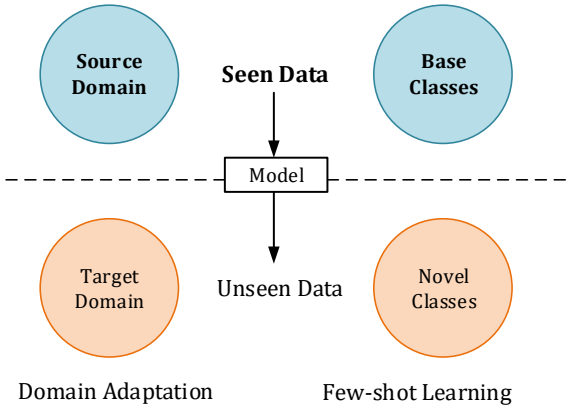


Fig. 3. Relationship between domain adaptation and few-shot learning. Both aim to recognize some unseen data with knowledge learned from a large amount of seen data

$C_{train}$ ($C_{train} \cap C_{test} = \emptyset$). Towards this goal, the samples of few-shot learning are typically divided into two sets, namely a support set $S_u$ (we added a subscript $u$ to differentiate it from above notation of source domain $S$) and a query set $Q$. Given a query image $I_q$ from $Q$, the objective of few-shot semantic segmentation is to segment the objects of $C_{test}$ in $I_q$ with only a few labeled images from the support set.

Since a query image may contain more than one class of $C_{test}$, existing few-shot semantic segmentation tasks can be generalized to a $N$ way $K$ shot segmentation problem where $N$ represents the number of classes that are to be simultaneously recognized in a single segmentation process, and $K$ is the number of labeled images of each class of $N$.

Note that although the labeled samples for each test class should be limited to a small number $K$, for a $K$-shot segmentation, the number of labeled samples for the base classes are usually available in large-scale to facilitate model training. Thus, the $K$-shot labeled samples are usually randomly sampled from the training set in each training episode. Suppose that a training batch contains $N$ classes that belong to $C_{train}$, we need to provide $N \times K$ support images for training. In other words, we need to theoretically process $B + NK$ ($B$ is the batch size) images in total for each training step, which can lead to a huge GPU memory requirement if $NK$ is large (e.g., $10 \times 5$). Moreover, $N$ may also differ for each training step. Therefore, current few-shot learning methods usually consider a binary segmentation (N=1) problem during training.

### C. Important Observation

Figure 3 illustrates the relationship between domain adaptation and few-shot learning tasks. It can be observed that both the domain adaptive semantic segmentation and few-shot semantic segmentation aim to recognize some unseen data with knowledge learned from seen data. In particular, domain adaptive semantic segmentation aims to recognize unseen data distributions while few-shot semantic segmentation aims to recognize unseen classes. Thus, if we view all the classes presented in unseen distribution in the target domain as unseen classes to the source domain, the domain adaptive semantic

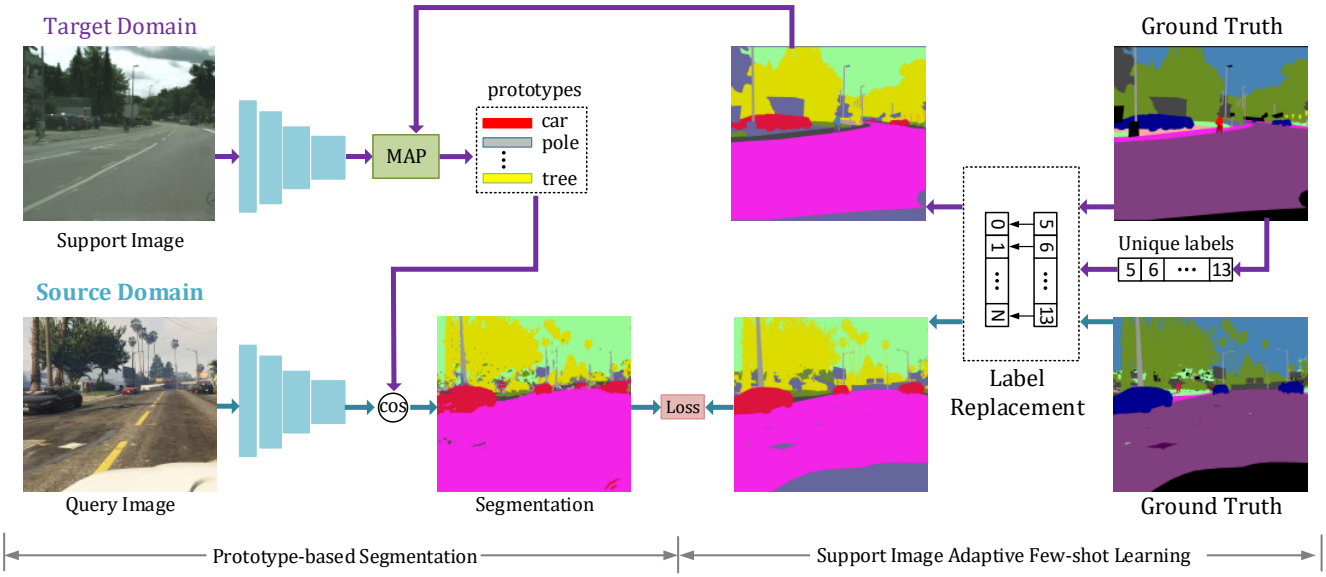


Fig. 4. Overview of our method. We propose to learn domain-invariant prototypes with a recent popular prototype-based few-shot learning framework. During training, we adopt few-shot annotated target images as the support set and treat all source images as the query set. In each training step, our model first embeds the support image and query image into semantic features using a Siamese Network. Then, a masked average pooling (MAP) operation is applied to the feature maps of support image to obtain class prototypes. Finally, predictions over the pixels of query image are obtained by finding the nearest class prototype to each pixel. To extend the prototype-based few-shot segmentation to high-resolution images containing arbitrary number of classes, we propose a support image adaptive training strategy in which the classes to be recognized as well as the number of support samples of them are fully determined by current support image. For segmentation of a test image, we need only perform prototype-based segmentation with all the class prototypes that are pre-computed from these few-shot annotated target images.

segmentation can be naturally formulated into a few-shot semantic segmentation problem.

## IV. Method

### A. Framework Overview

Figure 4 shows an overview of our method. Based on our hypothesis that humans perform object recognition via "similarity comparisons", we propose to predict the pixel labels by comparing its features to a number of domain-invariant class prototypes.

To learn domain-invariant prototypes, we collect a few annotated target images as the support set and treat all source images as the query set, following the common setting used for few-shot segmentation. For each training step, our model first embeds the support image (target domain) and query image (source domain) into semantic features using a Siamese Network. Then, a masked average pooling operation is applied to the feature maps of support image to obtain class prototypes. Finally, predictions over the pixels of query image are obtained by finding the nearest class prototype to each pixel and assigning it the label of that prototype.

We use VGG-16 [41] and adapt it into the DeepLab architecture for feature extraction. Specifically, we use the 5 convolutional blocks of VGG-16 as the backbone and remove the max pooling layer behind block 3 and block 4. To ensure the receptive field remains unchanged, the convolutions in block 4 and block 5 are replaced by dilated or atrous convolutions [42] with dilation rates set to 2 and 4, respectively.

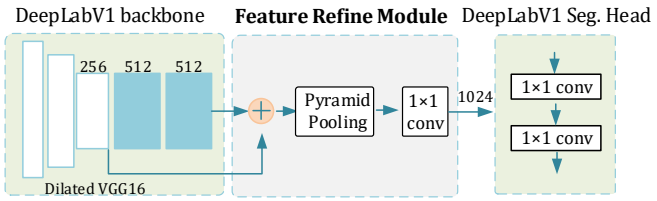


Fig. 5. Architecture of Feature Refinement Module (FRM).

### B. Feature Refinement Module

Intuitively, domain-invariant information is generally presented in high-level semantic concepts. For example, the semantic class label that we want the computer to recognize can be viewed as the highest-level domain-invariant information. Building on this insight, we further develop a feature refinement module (FRM) to encode as much semantic information as possible.

As shown in the Fig. 5, our FRM first concatenates the outputs of block 3 and block 5 of the backbone i.e., dilated VGG16. Next, we perform feature pyramid pooling, introduced in [43], to exploit context information. Finally, we adopt $1 \times 1$ convolution layers to transform the features to a lower dimensional (i.e. 1024) space.

### C. Construction of Support Set

For a $K$-shot semantic segmentation task, the $K$-shot labeled samples for training, i.e., the support set of each training episode, are usually randomly sampled from a large-scale dataset of base classes. However, since the support images for

our training are sampled from target domain in which labels are difficult to obtain, we must use a support set as small as possible. Below, we explain how we collect the support images from Cityscapes dataset [4] that we used as target domain in this paper.

As shown in Algorithm 1, we first define an array of size $N_{class}$ (e.g., 19 in Cityscapes) with all elements initialized to zero. This array will be used to accumulate the number of occurrences of the 19 classes of Cityscapes during dataset collection. Next, we randomly sample an image from the training set ($N_{image}$) of Cityscapes and add the corresponding entries of the above array with 1 if some classes are found in this image. Such accumulation of a specific class is ended if the occurrence of this class reaches to a predefined number $K_{shot}$ (similar to the $K$-shot). Only images that contribute novel accumulation to any class are then added to the support set. With this method, we finally collect 33 image-label pairs from the training set of Cityscapes for the support set by setting $K_{shot} = 5$. For convenience, we will call this dataset Cityscapes-5Shot in the rest of the paper.

Note that the target domain is typically unlabeled in practice, which means we need to manually annotate a number of target images for support set construction in this case. To minimize the time costs on human-annotation, the number of annotation samples for each class can be provided at instance-level rather than image-level.

**Algorithm 1** Construction of Support Set

**Require:** $N_{image}, K_{Shot}, N_{class}$
  $C_{occur} \leftarrow$ create an array of zeros of size $N_{class}$
  $I_{list} \leftarrow$ create an empty image list
  **for** $i = 0 : 1 : N_{image}$ **do**
    $l_i \leftarrow$ get the unique labels contained in $i$
    $n_i \leftarrow$ get the image name of $i$
    **for** $c = 0 : 1 : N_{class}$ **do**
      **if** $C_{occur}[c] < K_{Shot}$ & $c \in l_i$ **then**
        $C_{occur}[c] += 1$
        **if** $n_i \notin I_{list}$ **then**
          $I_{list} \leftarrow$ add $n_i$ to the $I_{list}$
        **end if**
      **end if**
    **end for**
  **end for**
  Return ($I_{list}$)

### D. Prototype based Semantic Segmentation

For a $N$-class semantic segmentation problem, the output at each pixel is a vector that consists of the probability of each class, which is usually computed by a softmax function as

$$b_c^{(h,w)} = \frac{\exp(\mathsf{w}_c^T x_{h,w})}{\sum_j^N \exp(\mathsf{w}_j^T x_{h,w})}, \tag{1}$$

where $x_{h,w}$ is the feature vector of pixel at spatial location $(h, w)$ and $\mathsf{w}_j$ (we use a different font here to differentiate it from the $w$ used for location denotation) contains the classification parameters of class $j$. Then, the training loss over semantic segmentation is usually computed by averaging the cross entropy at each pixel:

$$\mathcal{L}_{seg} = \frac{1}{\mathcal{HW}} \sum_{h,w} -\log\ b_{\hat{c}}^{(h,w)}, \tag{2}$$

where $\mathcal{HW}$ denotes the size of input image, $\hat{c}$ denotes the ground truth label.

Note that the value of $\mathsf{w}_c^T x_{h,w}$ in Eq. (1) is also called the score of class $c$. It essentially computes the distance between $x_{h,w}$ and $\mathsf{w}_c$ as an inner product. Since the $\mathsf{w}_c$ is fixed after model training, it can be viewed as a special prototype to some extent. Inspired by this observation, we propose to transform Eq. (1) to a general prototype based version as

$$b_c^{(h,w)} = \frac{\exp\left(\xi(p_c, x_{h,w})\right)}{\sum_j^N \exp\left(\xi(p_j, x_{h,w})\right)}, \tag{3}$$

where $p_c$ is the domain-invariant prototype of class $c$, and $\xi$ is a similarity function. In this paper, we adopt the commonly used cosine similarity function

$$\xi(x_1, x_2) = \frac{x_1 \cdot x_2}{\max(\|x_1\|_2 \cdot \|x_2\|_2, \epsilon)}. \tag{4}$$

Note that Eq. (3) requires the computation of the feature similarities with respect to $N$ class prototypes. We propose to compute these $N$ class prototypes by first computing prototypes separately from each image in Cityscapes-5shot dataset and then performing an averaging operation. Let $(I^{(j)}, Y^{(j)})$ be an image-label pair from the Cityscapes-5shot, our prototype extraction can be formally written as

$$p_c = \frac{\sum_{j=1}^K p_{c,j} \mathbb{1}[c \in \text{set}(Y^{(j)})]}{\sum_{j=1}^K \mathbb{1}[c \in \text{set}(Y^{(j)})]}, \tag{5}$$

where $K$ is the total number of support images, $\mathbb{1}[]$ is an indicator function that outputs 1 if the argument is true and 0 otherwise, set() is a function that outputs the unique labels contained in a labeled image, and $p_{c,j}$ is the prototype of class $c$ extracted from $I^{(j)}$ with the masked average pooling function

$$p_{c,j} = \frac{\sum_{h,w} x_{h,w}^{(j)} \mathbb{1}[Y_{h,w}^{(j)} = c]}{\sum_{h,w} \mathbb{1}[Y_{h,w}^{(j)} = c]}. \tag{6}$$

For the test stage, given the trained model, we can pre-compute the $p_c$ for all classes with Eq. (5) from the few-shot target images and save them to a local file. Then, we need only read these prototypes and applied them to Eq. (3) to perform semantic segmentation over test images.

For the training stage, however, $p_c$ need to be updated in each step, which could lead to a huge requirement on GPU memory if we consider the same number of classes as in testing/inference. In particular, we need to access all support images of the considered classes to extract the prototypes for each. The commonly used binary segmentation strategy (Section III-B) provides a solution to this problem, however, as detailed in Section I, that training strategy is not suitable for high-resolution images containing arbitrary number of classes. Next, we will detail our support image adaptive training strategy for this problem.

### E. Support Image Adaptive Training

Given a support image from target domain and a query image from source domain, we propose to use the classes contained in the support image for model training. More formally, denoting the unique labels contained in the support image $I_{st}$ and the query image $I_q$ by a label set $C_{st}$ and $C_q$, respectively, we recognize only classes that belong to the $C_{st}$ in each training episode. Thus, supposing the number of elements of $C_{st}$ is $N_{class}$, we essentially consider a $N_{class}-$way few-shot segmentation problem. We note that $N_{class}$ could be different in different training steps. Moreover, the label sequence sorted from $C_{st}$ may be nonconsecutive (e.g, $C_{st} = \{1, 7\}$), making it difficult to apply the commonly used cross entropy loss. To this end, we propose to replace the labels listed in $C_{st}$ with their entry indices, and all other labels with an invalid label 255. Given the above example $C_{st} = \{1, 7\}$, all pixels that belong to class 1 and 7 will be labeled 0 and 1, respectively. For the sake of clarity, we summarize our training sample generation in Algorithm 2, where the $G_{st}$ and $G_q$ represent the ground truth label image of support image $I_{st}$ and query image $I_q$, respectively.

Notice that our few-shot learning framework does not require that each prototype class must be provided with $K-$shot samples during training. Instead, the number of classes and their corresponding support samples are determined only by the current support image that is randomly sampled from the support set. This is reasonable since the number of annotated samples $K$ is not a constraint on the training stage but rather the test stage. Compared to the common binary segmentation based training, such a strategy has two important advantages. Firstly, it significantly reduces the number of support images required during training, resulting in a significant reduction in GPU memory consumption. Secondly, since we consider only the classes after sampling, our method can be applied to images without any constraints on their resolution or the number of objects contained.

**Algorithm 2** Support Image Adaptive Training

**Require:** $I_{st}, G_{st}, I_q, G_q$
  $C_{st} \leftarrow$ get the unique labels contained in $G_{st}$
  $C_{st} \leftarrow$ delete the invalid label in $C_{st}$
  $C_q \leftarrow$ get the unique labels contained in $G_q$
  **for** $i \in C_q$ **do**
    **if** $i \notin C_{st}$ **then**
      $G_q \leftarrow$ replace label $i$ in $G_q$ with the invalid label
    **end if**
  **end for**
  **for** $i \in C_{st}$ **do**
    $e_i \leftarrow$ get the entry index of $i$ in $C_{st}$
    $G_{st}, G_q \leftarrow$ replace label $i$ in $G_{st}$ and $G_q$ with $e_i$
  **end for**
  Return $I_{st}, G_{st}, I_q, G_q$

### F. Loss Function

Similar to most prototype-based few-shot segmentation methods, our basic few-shot domain adaptation framework requires only the annotations of support images for prototype extraction and the prototype-based segmentation is supervised by the dense annotations of query images. While the support image and query image in our method come from different domains and our goal is to learn domain-invariant prototypes, we further apply the annotations of support images to supervise the prototype-based segmentation on themselves. Thus, the final loss for our model training is composed of two parts: the segmentation loss on query images $\mathcal{L}_{seg}^{Q}$ and the segmentation loss on support images $\mathcal{L}_{seg}^{ST}$. Both are obtained by computing the cross entropy loss via Eq. (2). Given that the support images are few, we treat the segmentation loss on support images as an auxiliary loss and multiply it with a balancing factor $\alpha$ to avoid overfitting to the few support images. Our final loss can be formally written as

$$\mathcal{L} = \mathcal{L}_{seg}^{Q} + \alpha \mathcal{L}_{seg}^{ST}. \tag{7}$$

## V. Experiments

Following common practice in domain adaptive semantic segmentation research, we first validate the effectiveness of our method by conducting experiments on GTA5-to-Cityscapes adaptation task. The Cityscapes [4] dataset contains 2,975, 500, and 1,525 densely annotated urban images for training, validation, and testing, respectively. The images are collected from 50 European cities. Note that the annotations of 1,525 test images are not publicly available. The GTA5 dataset [44] contains 24,966 synthetic images from a well-known computer game named Grand Theft Auto. It shares 19 common labels with the Cityscapes dataset.

### A. Implementation Details

Since the GTA5 consists of 24,966 images with resolution up to $1914 \times 1052$, training over the GTA5 is usually time-consuming. Hence, the GTA5 contributors divided the data into 10 parts. For our ablation studies, we use only the first part which contains 2,500 images. After that, we train the final model over the entire GTA5 dataset for comparison to state-of-the-art methods.

We initialized the VGG16 backbone with ImageNet pre-trained weights. Next, we divided the VGG16 into five blocks according to the feature resolution and fixed the weights of the first three blocks when adapting it to the segmentation task. For convenience, we will refer to our few-shot domain adaptation method as FSDA in the rest of the paper.

### B. Experimental Settings

We implemented and trained our deep model in PyTorch [45]. We trained our model for 50 epochs on $1024\times1024$ images (batch size 1) using the SGD optimizer with a base learning rate of 0.001 and momentum 0.9. The learning rate was updated with the commonly used "poly" policy [42]. For data augmentation, we used random scaling (0.9 to 1.1), random horizontal flip and random rotation ($-10^\circ$ to $10^\circ$). Our training batch sampling follows the commonly used "scaling-and-cropping" strategy which first resizes a training image to

TABLE I
PERFORMANCE OF OUR BASELINE ON GTA5-TO-CITYSCAPES ADAPTATION. NOTE THAT WE USE ONLY $1/10$ SOURCE IMAGES IN THIS EXPERIMENT. THE "STLOSS" REPRESENTS THE AUXILIARY SEGMENTATION LOSS ON SUPPORT IMAGE, AND THE FOLLOWED NUMBER IS THE BALANCING FACTOR .

| Model | road | side. | build. | wall | fence | pole | light | sign | vege. | terr. | sky | person | rider | car | truck | bus | train | motorcy. | bike | **mIoU** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Only | 60.3 | 21.2 | 61.9 | 3.7 | 13.1 | 21.6 | 20.7 | 5.7 | 77.3 | 25.6 | 22.7 | 32.2 | 0.0 | 40.3 | 5.2 | 1.7 | 0.0 | 0.0 | 0.0 | **21.8** |
| Few-shot Only (5 shot) | 88.4 | 42.5 | 76.8 | 9.9 | 10.8 | 12.7 | 0.3 | 27.7 | 83.5 | 26.6 | 68.2 | 41.6 | 0.0 | 71.6 | 3.8 | 2.1 | 0.0 | 0.0 | 26.2 | **31.2** |
| Few-shot Only (1 shot) | 85.1 | 24.7 | 68.8 | 4.2 | 0.0 | 0.0 | 0.0 | 0.0 | 82.7 | 16.6 | 41.8 | 16.2 | 0.0 | 55.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | **20.9** |
| FSDA | 82.8 | 37.6 | 69.3 | 13.6 | 15.2 | 23.3 | 12.3 | 31.6 | 78.3 | 22.9 | 70.5 | 38.6 | 5.7 | 63.6 | 4.4 | 6.7 | 1.2 | 2.6 | 15.6 | **31.4** |
| FSDA+stloss (0.1) | 91.7 | 51.0 | 78.8 | 17.5 | 12.5 | 29.0 | 22.2 | 40.0 | 83.3 | 26.2 | 76.7 | 39.1 | 6.5 | 73.9 | 5.2 | 12.6 | 1.9 | 4.1 | 14.6 | **36.2** |
| FSDA+stloss (0.2) | 91.4 | 52.5 | 79.8 | 18.0 | 14.9 | 30.8 | 22.1 | 38.4 | 84.0 | 28.9 | 80.0 | 35.6 | 7.1 | 73.4 | 4.8 | 13.2 | 1.8 | 4.6 | 15.8 | **36.7** |
| FSDA+stloss (0.4) | 91.6 | 49.2 | 79.3 | 15.3 | 13.2 | 28.5 | 23.7 | 40.1 | 84.4 | 32.2 | 74.7 | 37.4 | 7.1 | 75.3 | 5.7 | 11.3 | 2.2 | 4.7 | 16.6 | **36.5** |
| FSDA+stloss (0.6) | 91.9 | 53.9 | 79.5 | 14.3 | 14.7 | 29.2 | 24.0 | 39.2 | 84.2 | 29.0 | 72.4 | 39.2 | 6.8 | 75.2 | 5.3 | 10.6 | 2.1 | 5.4 | 16.2 | **36.5** |
| FSDA (unseen)+stloss (0.2) | 92.6 | 54.0 | 79.0 | 14.5 | 16.3 | 29.2 | 26.1 | 40.9 | 84.3 | 29.3 | 80.4 | 34.2 | 8.9 | 72.4 | 7.2 | 7.1 | 2.6 | 4.2 | 17.0 | **36.9** |
| FSDA (1 shot) | 85.8 | 37.4 | 70.8 | 13.7 | 12.9 | 23.9 | 18.0 | 28.9 | 78.8 | 24.6 | 51.9 | 39.5 | 7.5 | 71.2 | 5.1 | 7.8 | 1.8 | 4.7 | 9.9 | **31.3** |
| FSDA (1 shot)+stloss (0.2) | 88.7 | 35.7 | 76.9 | 16.6 | 10.5 | 26.4 | 25.6 | 33.2 | 83.4 | 28.0 | 53.5 | 49.5 | 6.1 | 74.0 | 5.2 | 6.7 | 2.1 | 4.8 | 15.5 | **33.8** |

a random scale and then crops an image patch equal to the training size from the scaled image. All experiments were conducted on a computer equipped with a Titan X (12GB) GPU and 32G RAM. We adopt the commonly used mean intersection over union (mIoU) as the major metric for model comparison.

### C. Comparison to Baselines

We first implemented the most straightforward adaptation strategy, i.e., directly applying model trained on GTA5 to Cityscapes, as baseline and achieved 21.8% mIoU (see Table I). Since our method utilized few-shot annotated target images for domain adaptation, we further evaluated another baseline named few-shot only in which the model was trained on the few-shot (33) support images of Cityscapes. Our few-shot domain adaptation is trained for 50 epochs in the following experiments. Each training image is associated with a support image from the 33 few-shot images in this case. In other words, the few-shot training set would be traversed for $(50 \times 2500)/33 \approx 3788$ epochs in our FSDA framework. Thus, for a fair comparison, we should train the model for 3788 epochs for the few-shot baseline. However, the trained model may easily overfit to the few-shot training images given such a large number of training epochs. On the other hand, we found that the few-shot baseline starts to converge at about 180 epochs. Thus, we set the training epochs to 200 for the few-shot baseline, which yielded 31.2% mIoU on the validation set of Cityscapes.

As shown in Table I, compared to using the source only, our FSDA framework obtained a performance gain of 9.6 percentage points (31.4% mIoU), showing the effectiveness of our few-shot adaptation framework. Compared to the few-shot only baseline, our FSDA obtained only 0.2 percentage point performance gain. Nevertheless, our method can segment classes (e.g., rider, train and motorcycle) that are generally missed by the few-shot only baseline. Later in Section V-F, we show that our method can boost this gain significantly when fewer support images are used. Moreover, we used these few-shot annotations only for prototype extraction in our basic FSDA framework. Our following experiments show that we can obtain more gains when these annotations are further used to directly supervise the segmentation model.

### D. Ablation Study on Loss

Our basic FSDA method employed only $\mathcal{L}_{seg}^{Q}$ for model training. To validate how the auxiliary segmentation loss $\mathcal{L}_{seg}^{ST}$ in Eq. (7) influences the training, we tested different values of the balancing factor.

We first tested a small balancing factor of 0.1 and achieved 36.2% mIoU (see Table I), obtaining a performance gain of 4.8% mIoU over our basic FSDA framework. The performance further improved to 36.7% mIoU when the balancing factor was increased to 0.2. Since these performances are on the target domain, this can be well explained because $\mathcal{L}_{seg}^{ST}$ directly utilized the supervision signal from the target domain. However, we did not obtain further gain but rather a reduction in mIoU when we further increased the balancing factor, i.e., letting the supervision from target domain play a much greater role in model training. In particular, the performance in terms of mIoU decreased from 36.7% to 36.5% when the balancing factor was increased from 0.2 to 0.4. This is mainly because the annotated target images we used were very few. Thus, increasing the weight of $\mathcal{L}_{seg}^{ST}$ would also promote the prototype learning to overfit to these few-shot annotations.

With our best performing FSDA framework, we show qualitative comparisons between our method and baselines in Fig. 6. It can be observed that the source only baseline has a bias towards stuff classes that usually occupy a large continuous area such as wall and road classes. For example, almost the whole car (center-right) in the first image and the whole bus in the fifth image are classified into wall by the source only baseline. This bias was greatly reduced by our framework. Compared with the few-shot baseline, our method performed much better on small objects such as traffic light, traffic sign,

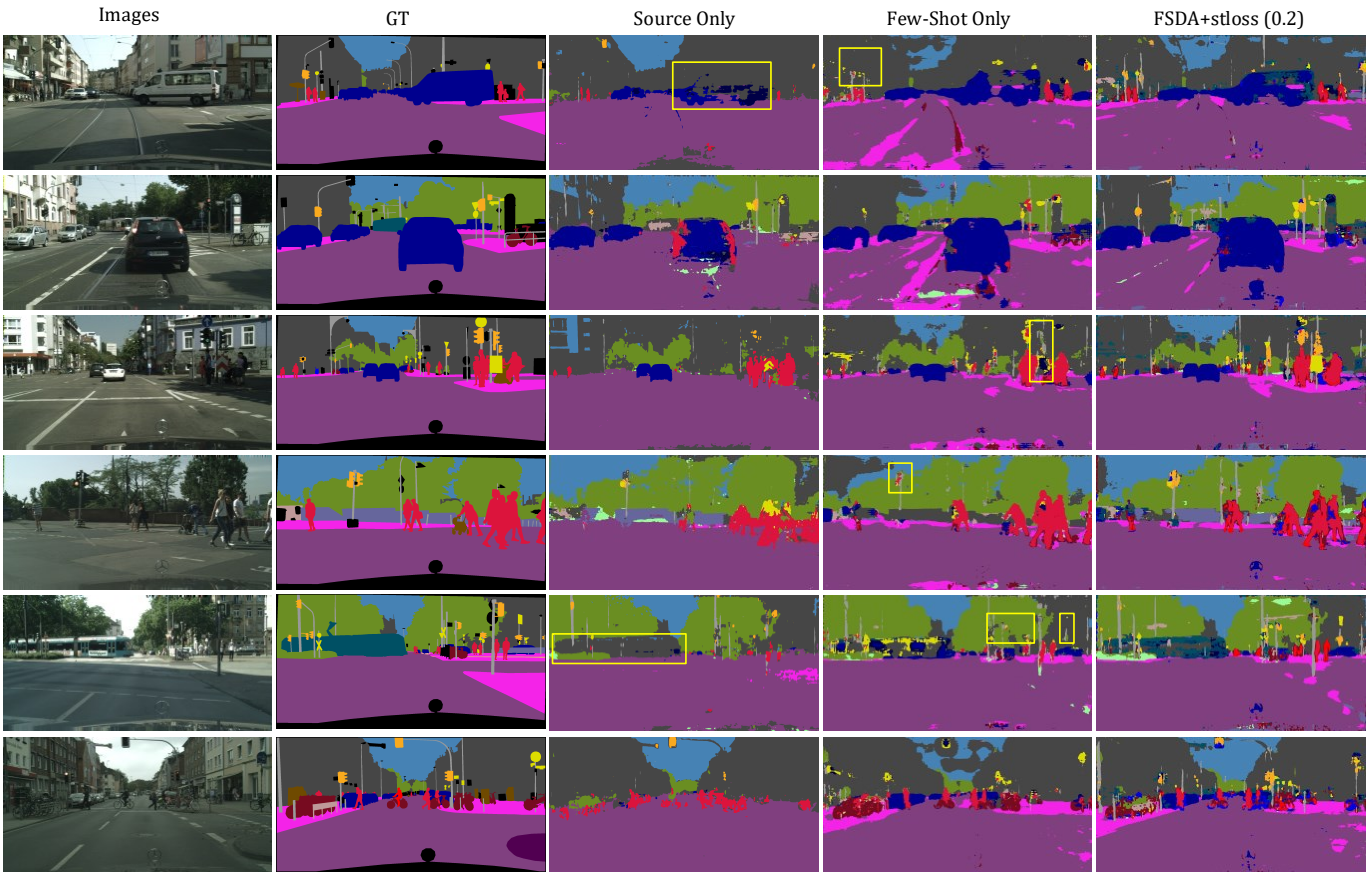


Fig. 6. Qualitative results on GTA5-to-Cityscapes adaptation. The source only baseline has a bias towards stuff classes that occupy a large continuous area, while the few-shot only baseline cannot recognize small objects well due to the limitation on the total number of training pixels.

and pole. This is mainly because pixel-wise loss based training can be easily dominated by large objects especially when the total training samples are few.

### *E. Generalization to Unseen Support Image*

We used the same support images for training and testing in the above experiments. In this experiment, we generate three groups of support images that are different from those 33 images used for training, which can be easily implemented by applying a random shuffle operation with different seeds to the training set of Cityscapes before employing Algorithm 1. However, this is not recommended as this would increase the requirement for annotations on the target domain. The main purpose of this experiment is to simply verify whether our FSDA can generalize to unseen support images. Thus, these three groups support images were only used for testing. As shown in the third last row of Table I, our FSDA achieved an average 36.9% mIoU with these three group unseen support images, which is similar to the performance achieved with the seen support images.

### *F. One Shot Domain adaptation*

We used the Cityscapes-5Shot dataset as the support set in the above experiments. To further validate whether our method can work with fewer labeled target images, we considered a one-shot domain adaptation problem. To this end, we generated a Cityscapes-1Shot dataset that contains only 7 annotated images for training using Algorithm 1.

Similarly, we first implemented the few-shot only baseline with Cityscapes-1Shot and achieved 20.9% mIoU i.e. a 10.3 percentage point drop compared with using Cityscapes-5Shot. This is mainly due to the difficultly of learning effective features for "things" objects (e.g. car, pole) from only one-shot samples, especially when "stuff" objects (e.g. road) occupy larger proportions of the images. As shown in Table I, almost all "things" objects except the car could not be correctly segmented by the few-shot only baseline when only one-shot support images were used. In contrast, our method still achieved a relatively steady performance (33.8% mIoU) compared with the 36.7% mIoU achieved in 5-shot experiments. Moreover, our method outperformed the source only baseline and few-shot only baseline by 12 and 12.9 percentage points, respectively.

### *G. Comparison With State-of-The-Art*

For comparison to representative methods published in recent years, we trained our FSDA on the whole GTA5 dataset for 10 epochs using the same training policies listed in Section V-B. Table II shows detailed comparative results. The proposed FSDA achieved 39.4% mIoU on Cityscapes validation set outperforming many methods (e.g., FCN wild, CDA,

TABLE II
PERFORMANCE COMPARISON WITH STATE-OF-THE-ARTS ON GTA5-TO-CITYSCAPES ADAPTATION.

| Model | road | side. | build. | wall | fence | pole | light | sign | vege. | terr. | sky | person | rider | car | truck | bus | train | motorcy. | bike | **mIoU** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FCN wild [46] | 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 |
| CDA [47] | 74.9 | 22.0 | 71.7 | 6.0 | 11.9 | 8.4 | 16.3 | 11.1 | 75.7 | 13.3 | 66.5 | 38.0 | 9.3 | 55.2 | 18.8 | 18.9 | 0.0 | 16.8 | 14.6 | 28.9 |
| MCD [48] | 86.4 | 8.5 | 76.1 | 18.6 | 9.7 | 14.9 | 7.8 | 0.6 | 82.8 | 32.7 | 71.4 | 25.2 | 1.1 | 76.3 | 16.1 | 17.1 | 1.4 | 0.2 | 0.0 | 28.8 |
| I2I [49] | 85.3 | 38.0 | 71.3 | 18.6 | 16.0 | 18.7 | 12.0 | 4.5 | 72.0 | **43.4** | 63.7 | 43.1 | 3.3 | 76.7 | 14.4 | 12.8 | 0.3 | 9.8 | 0.6 | 31.8 |
| CyCADA [50] | 85.2 | 37.2 | 76.5 | 21.8 | 15.0 | 23.8 | 22.9 | 21.5 | 80.5 | 31.3 | 60.7 | 50.5 | 9.0 | 76.9 | 17.1 | 28.2 | 4.5 | 9.8 | 0.0 | 35.4 |
| CBST [17] | 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 |
| Advent [51] | 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 |
| PyCDA [52] | 86.7 | 24.8 | 80.9 | 21.4 | 27.3 | 30.2 | 26.6 | 21.1 | **86.6** | 28.9 | 58.8 | 53.2 | 17.9 | 80.4 | 18.8 | 22.4 | 4.1 | 9.7 | 6.2 | 37.2 |
| DPR [53] | 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 |
| CLAN [54] | 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 |
| PIT [55] | 86.2 | 35.0 | 82.1 | 31.1 | 22.1 | 23.2 | 29.4 | 28.5 | 79.3 | 31.8 | 81.9 | 52.1 | 23.2 | 80.4 | 29.5 | 26.9 | 30.7 | 20.5 | 1.2 | 41.8 |
| FDA [56] | 86.1 | 35.1 | 80.6 | 30.8 | 20.4 | 27.5 | 30.0 | 26.0 | 82.1 | 30.3 | 73.6 | 52.5 | 21.7 | 81.7 | 24.0 | 30.5 | 29.9 | 14.6 | 24.0 | 42.2 |
| FADA [57] | **92.3** | 51.1 | 83.7 | 33.1 | 29.1 | 28.5 | 28.0 | 21.0 | 82.6 | 32.6 | 85.3 | 55.2 | **28.8** | 83.5 | 24.4 | 37.4 | 0.0 | 21.1 | 15.2 | 43.8 |
| LDR [58] | 90.1 | 41.2 | 82.2 | 30.3 | 21.3 | 18.3 | 33.5 | 23.0 | 84.1 | 37.5 | 81.4 | 54.2 | 24.3 | 83.0 | 27.6 | 32.0 | 8.1 | **29.7** | 26.9 | 43.6 |
| CD-AM [26] | 90.1 | 46.7 | 82.7 | **34.2** | 25.3 | 21.3 | 33.0 | 22.0 | 84.4 | 41.4 | 78.9 | 55.5 | 25.8 | 83.1 | 24.9 | 31.4 | 20.6 | 25.2 | 27.8 | 44.9 |
| SA-I2I [59] | 91.1 | 46.4 | 82.9 | 33.2 | 27.9 | 20.6 | 29.0 | 28.2 | 84.5 | 40.9 | 82.3 | 52.4 | 24.4 | 81.2 | 21.8 | 44.8 | **31.5** | 26.5 | 33.7 | 46.5 |
| SAC [10] | 90.0 | **53.1** | **86.2** | 33.8 | **32.7** | **38.2** | **46.0** | 40.3 | 84.2 | 26.4 | **88.4** | **65.8** | 28.0 | **85.6** | **40.6** | **52.9** | 17.3 | 13.7 | 23.8 | **49.9** |
| Ours (1shot) | 88.9 | 39.0 | 78.9 | 19.2 | 14.9 | 27.1 | 24.7 | 35.4 | 84.6 | 34.5 | 66.4 | 51.5 | 4.2 | 77.6 | 5.3 | 3.5 | 1.7 | 5.0 | 8.0 | **35.3** |
| Ours (5shot) | 90.6 | 47.0 | 80.1 | 15.4 | 15.6 | 29.3 | 30.7 | **46.1** | 82.4 | 26.4 | 76.4 | 51.9 | 12.3 | 76.3 | 7.1 | 9.7 | 1.2 | 7.6 | **42.0** | 39.4 |

MCD) that were published before 2020. When compared with state-of-the-art results obtained in past two years, our method lost its competitiveness. For example, the SAC method achieved close to 50% mIoU with VGG16. Nevertheless, it is noteworthy that the main purpose of this paper is not to improve the state-of-the-art performance achieved by current self-training based methods and adversarial training based methods, but rather to provide a simple generic alternative for the domain adaptation task. Our one-stage training method is much easier to implement compared to existing methods since it does not require either multiple rounds self-training on large-scale target images or optimizing adversarial losses. Moreover, while these compared methods only focus on a standard domain adaptation task, our method can be generalized to many variants of both domain adaptation task and few-shot learning tasks as follows.

**Open Set Domain adaptation (OSDA)**: Denoting the label set considered in source domain and target domain by $C_S$ and $C_T$, respectively, there are many possible variants for domain adaptation. Most existing methods consider a standard domain adaptation problem where $C_S = C_T$. However, less attention has been paid to the OSDA problem, where $C_S \neq C_T \ \& \ C_S \cap C_T = C_{com} \neq \emptyset$, i.e. there are a few private classes in target domain or both. It can be observed that the OSDA needs to recognize some novel classes in target domain, which shares a common character with the few-shot learning task. In other words, the OSDA can be naturally extended to a few-shot learning task with domain shifts if some annotated samples are available for each private class of the target domain. Thus, our FSDA framework, developed from prototype-based few-shot learning, can also be easily applied to solve the OSDA problem. In particular, we can use classes that are contained in both the target and source domain, i.e., $C_{com}$, as the base classes to train the network for domain-invariant prototype extraction. Then, for both base classes and novel classes in target domain, we need only perform prototype-based segmentation using Eq. (3) with class prototypes extracted from their few-shot annotated samples. We perform OSDA experiments in Section V-H.

**Multi-Source Domain adaptation (MSDA)**: The MSDA task considers a case that there are multiple source domains presented in different distributions for knowledge transfer. Similar to the standard domain adaptation, the basic MSDA assumes that all domains share the same label set, i.e., $\forall i, C_S^{(i)} = C_T$, where $i$ is the index of source domain. The basic MSDA can be evolved into a multi-source open set domain adaptation (MS-OSDA) problem if there are a few classes contained only in the target domain. Since the core idea of our FSDA is to learn domain-invariant prototypes, i.e., we do not care about the domain-wise labels of images and need only to merge all source domains to a single domain for the basic MSDA problem. With this setting, the MS-OSDA problem can also be solved with the method for OSDA mentioned above.

**Generalized Few-Shot Learning:** Compared to state-of-the-art domain adaptation, another important advantage of our method is that it can also be used to solve a generalized few-shot learning problem without constraints on the image resolution and the number of classes (see Section IV-E).

### *H. SYNTHIA to Cityscapes*

In the above experiments, GTA5 was the source domain. To show that our method can be generalized to different

TABLE III
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART ON SYNTHIA-TO-CITYSCAPES ADAPTATION.

| Model | road | side. | build. | wall | fence | pole | light | sign | vege. | terr. | sky | person | rider | car | truck | bus | train | motorcy. | bike | **mIoU16** | **mIoU13** | **mIoU19** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CDA [47] | 65.2 | 26.1 | 74.9 | 0.1 | 0.5 | 10.7 | 3.7 | 3.0 | 76.1 | - | 70.6 | 47.1 | 8.2 | 43.2 | - | 20.7 | - | 0.7 | 13.1 | 28.9 | - | - |
| Advent [51] | 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 | - |
| DPR [53] | 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 | - |
| PyCDA [52] | 80.6 | 26.6 | 74.5 | 2.0 | 0.1 | 18.1 | 13.7 | 14.2 | 80.8 | - | 71.0 | 48.0 | 19.0 | 72.3 | - | 22.5 | - | 12.1 | 18.1 | 35.9 | - | - |
| CBST [17] | 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 | - | - |
| PIT [55] | 81.7 | 26.9 | 78.4 | 6.3 | 0.2 | 19.8 | 13.4 | 17.4 | 76.7 | - | 74.1 | 47.5 | 22.4 | 76.0 | - | 21.7 | - | **19.6** | 27.7 | 38.1 | - | - |
| FADA [57] | 80.4 | 35.9 | **80.9** | 2.5 | 0.3 | **30.4** | 7.9 | 22.3 | 81.8 | - | **83.6** | 48.9 | 16.8 | **77.7** | - | **31.1** | - | 13.5 | 17.9 | 39.5 | - | - |
| FDA [56] | 84.2 | 35.1 | 78.0 | 6.1 | 0.4 | 27.0 | 8.5 | 22.1 | 77.2 | - | 79.6 | 55.5 | 19.9 | 74.8 | - | 24.9 | - | 14.3 | 40.7 | 40.5 | - | - |
| CD-AM [26] | 73.0 | 31.1 | 77.1 | 0.2 | 0.5 | 27.0 | 11.3 | 27.4 | 81.2 | - | 81.0 | **59.0** | 25.6 | 75.0 | - | 26.3 | - | 10.1 | 47.4 | 40.8 | - | - |
| LDR [58] | 73.7 | 29.6 | 77.6 | 1.0 | 0.4 | 26.0 | 14.7 | 26.6 | 80.6 | - | 81.8 | 57.2 | **24.5** | 76.1 | - | 27.6 | - | 13.6 | 46.6 | 41.1 | - | - |
| SA-I2I [59] | 79.1 | 34.0 | 78.3 | 0.3 | 0.6 | 26.7 | 15.9 | 29.5 | 81.0 | - | 81.1 | 55.5 | 21.9 | 77.2 | - | 23.5 | - | 11.8 | 47.5 | 41.5 | - | - |
| CLAN [54] | 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 | - |
| Source Only | 2.6 | 16.3 | 33.8 | 0.3 | 0.0 | 18.7 | 2.4 | 4.7 | 71.2 | - | 33.0 | 41.3 | 1.1 | 40.7 | - | 7.5 | - | 0.3 | 2.1 | **17.2** | 19.7 | 14.5 |
| Ours (1shot) | 86.0 | 32.7 | 75.9 | 7.4 | 0.7 | 28.0 | 17.5 | 30.3 | 83.3 | 15.2 | 70.8 | 55.2 | 6.4 | 71.6 | 0.0 | 7.7 | 0.7 | 2.3 | 19.2 | 37.2 | 43.0 | 32.2 |
| Ours (5shot) | **89.3** | **43.4** | 79.1 | **15.4** | **9.9** | 29.3 | **24.3** | **34.8** | **83.5** | **28.8** | 79.8 | 56.5 | 12.0 | 75.3 | 1.6 | 13.6 | 0.2 | 9.4 | **47.7** | **43.9** | **49.9** | **38.6** |

datasets, we also conducted experiments on the SYNTHIA-to-Cityscapes adaptation task. Following common practice, we adopted the SYNTHIA-RAND-CITYSCAPES dataset [5] which contains 9,400 densely annotated images for our study. Different from GTA5, SYNTHIA shares only 16 common labels with the 19 classes present in the Cityscapes dataset. In other words, the SYNTHIA-to-Cityscapes adaptation is essentially an OSDA problem. Nevertheless, most existing methods consider a standard domain adaptation problem on this dataset and report mIoU on the 16 common classes. A few methods considered only 13 classes, where the wall, fence and pole were also excluded. In this experiment, we consider both the OSDA and standard case. For convenience, we denoted these mIoU values computed based on different class numbers by mIoU$N$ (e.g., mIoU16).

**Overall results**: Similar to the experiments on GTA5, we first trained the source only baseline for 50 epochs and achieved 17.2% mIoU16, 19.7% mIoU13, and 14.5% mIoU19. Then, we trained our FSDA using different number of support images for the same number of epochs. With 1-shot support images, we achieved 37.2% mIoU16, 43.0% mIoU13 and 32.2% mIoU19. With 5-shot support images, we achieved 43.9% mIoU16, 49.9%mIoU13, and 38.6% mIoU19.

**Quantitative Analysis:** We first analyze our results from the perspective of standard domain adaptation, i.e., considering the mIoU16 for analysis. It can be observed from Table III that our FSDA achieved the state-of-the-art performance on SYNTHIA-to-Cityscapes adaptation task. Moreover, we see that the 17.2% mIoU16 of source only baseline achieved with SYNTHIA lags far behind that obtained with GTA5 (i.e. 24.0% mIoU16 derived from the 21.8% mIoU19 in Table I), even though the number of source images of SYNTHIA is significantly larger than that of GTA5 (9,400 vs 2,500). This is mainly because many images of SYNTHIA were sampled from the same scene with different lighting or weather conditions, making the domain shift with respect to Cityscapes much larger than that of GTA5. As a result, the performance degradation is also visible in state-of-the-art domain adaptation methods when the source domain is changed to SYNTHIA from GTA5 (see Fig. 7). In particular, the performance in terms of mIoU16 of PyCDA, PIT, FDA, FADA, LDR and SA-I2I dropped by 5.0, 5.8, 4.4, 9.0, 6.1 and 7.8 percentage points, respectively. In contrast, our method suffered only a 0.7 percentage point drop in performance. This shows that our method is more robust to the data distribution of source domain.

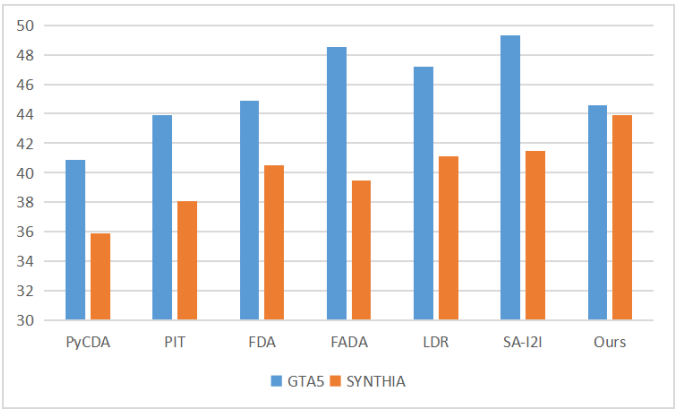


Fig. 7. Performance of state-of-the-art methods on Cityscapes using different source domains. Most methods encounter a significant performance drop when the source domain changes from GTA5 to SYNTHIA. In contrast, our method has only a minor performance drop.

From the perspective of OSDA, our FSDA achieved 28.8%, 1.6% and 0.2% mIoU on the three private classes of Cityscapes namely, terrain, truck and train. Although not very impressive, this is acceptable when compared to few-shot only baseline, which achieved 26.6%, 3.8% and 0.0% mIoU on these three classes (see Table I).

**Qualitative Analysis:** We also provide qualitative results for experiments on the SYNTHIA-to-Cityscapes adaptation task.

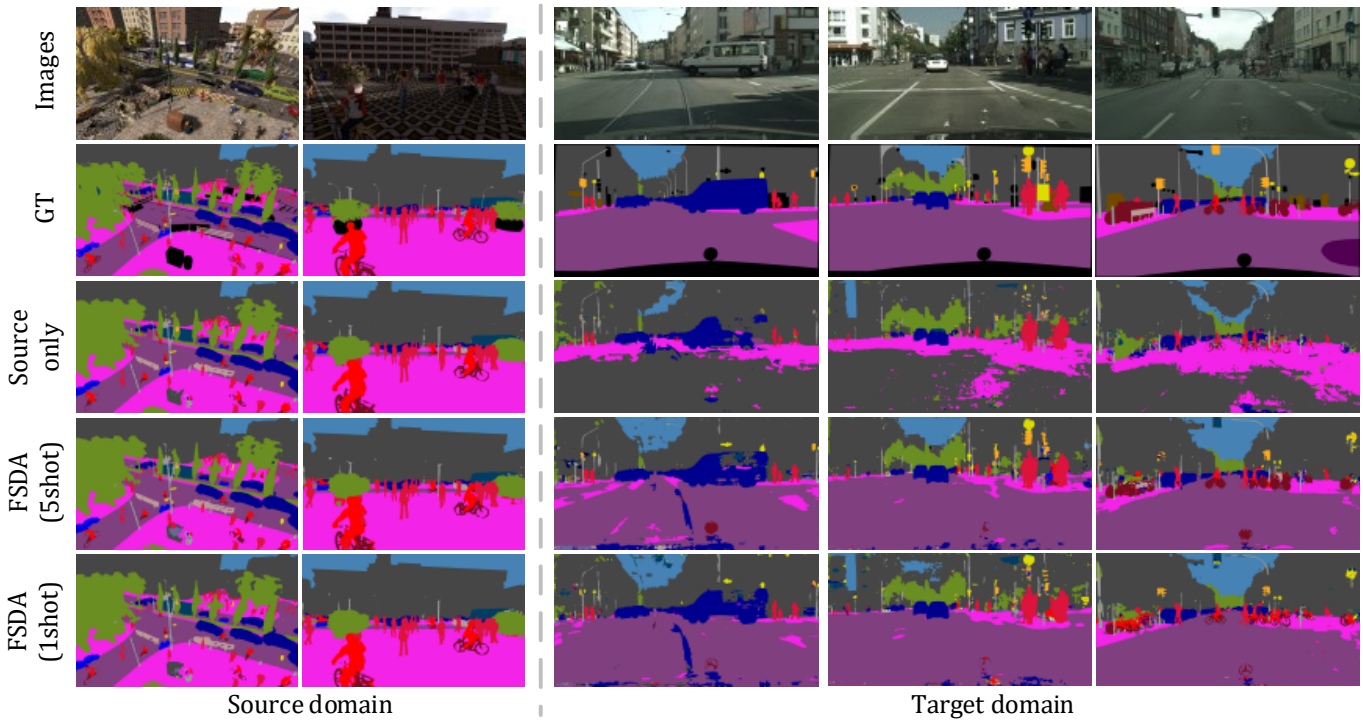


Fig. 8. Qualitative results on SYNTHIA-to-Cityscapes adaptation. The first two columns are the results on the source domain, and the last three columns are the results on the target domain.

As shown in Fig. 8, we applied the source only baseline and two variants of our method to SYNTHIA and Cityscapes dataset. Due to large distribution shifts, the source only baseline trained on SYNTHIA tends to segment large area of the road as building when adapted to the Cityscapes dataset. In contrast, our method can distinguish the road and building in most cases, and perform much better on many other objects like pole, traffic sign, and traffic light. In addition, the results on SYNTHIA images show that our method can achieve almost the same performance with the source only baseline, indicating that our method can reserve its representation ability on source domain after the domain adaptation, which is useful when it is required to applying the same model to both the source and target domains.

## VI. Conclusion and Future Work

In this paper, we introduced a simple generic framework for domain adaptation called FSDA. We showed that the FSDA can be generalized to many variants of domain adaptation tasks such as OSDA and MSDA with the support of a few annotated images from the target domain. In addition, our FSDA, developed from prototype-based few-shot learning, also provides a solution to few-shot learning on images that contain arbitrary number of objects. The proposed FSDA achieved competitive performance (39.4% mIoU) on the GTA5-to-Cityscapes adaptation task, and state-of-the-art performance (43.9% mIoU16) on the SYNTHIA-to-Cityscapes adaptation task. To the best of our knowledge, our method is the first to unify the domain adaptation and few-shot learning into a single framework.

Since the prototype-based method we used has been proven to be effective in few-shot learning, this paper mainly focused on domain adaptation task while paying less attention to apply the unified framework to few-shot segmentation. Nevertheless, current few-shot segmentation methods are still limited to images of low-resolution and containing only a few objects. Thus, an interesting direction is to extend our method to few-shot segmentation on images containing arbitrary number of classes. Many novel strategies (e.g., adaptive prototype learning [12], self-guided learning [60]) have been proposed to improve the performance of prototype-based few-shot segmentation in the past several years. Hence, another promising future work is to validate whether such strategies can yield better domain-invariant prototypes when incorporated into our framework.

We have also tried the ResNet [61] as the backbone in our experiments. However, it did not yield better results compared with using VGG. We hypothesize this mainly caused by the batch normalization (BN) [62] layer used in ResNet. The BN essentially computes the statistical properties of the features during the entire training process. Compared with existing methods that adopt all target images for training, our method access only a few target images in the training process. Thus, it is difficult for the BN layer to learn effective feature statistics of the target domain. To solve this problem, a future direction can be to perform unsupervised feature learning on target domain before applying our method.

## References

[1] Z. Yang, H. Yu, M. Feng, W. Sun, X. Lin, M. Sun, Z.-H. Mao, and A. Mian, "Small object augmentation of urban scenes for real-time semantic segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 5175–5190, 2020.

[2] Z. Yang, H. Yu, Q. Fu, W. Sun, W. Jia, M. Sun, and Z.-H. Mao, "Ndnet: Narrow while deep network for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5508–5519, 2020.

[3] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, "Effective use of synthetic data for urban scene semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 84–100.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[5] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[6] G. Li, G. Kang, W. Liu, Y. Wei, and Y. Yang, "Content-consistent matching for domain adaptive semantic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 440–456.

[7] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.

[8] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 414–12 424.

[9] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9092–9101.

[10] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 384–15 394.

[11] M. Siam, B. N. Oreshkin, and M. Jagersand, "Amp: Adaptive masked proxies for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5249–5258.

[12] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8334–8343.

[13] W. Liu, C. Zhang, G. Lin, and F. Liu, "Crnet: Cross-reference networks for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4165–4173.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[15] H. Ma, X. Lin, Z. Wu, and Y. Yu, "Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4051–4060.

[16] F. Fleuret *et al.*, "Uncertainty reduction for model adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9613–9623.

[17] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 289–305.

[18] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2090–2099.

[19] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5982–5991.

[20] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7472–7481.

[21] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1335–1344.

[22] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 982–991.

[23] Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 518–534.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[25] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, "All about structure: Adapting structural information across domains for boosting semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1900–1909.

[26] J. Yang, W. An, C. Yan, P. Zhao, and J. Huang, "Context-aware domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 514–524.

[27] Y. Yang, D. Lao, G. Sundaramoorthi, and S. Soatto, "Phase consistent ecological domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9011–9020.

[28] X. Yue, Z. Zheng, S. Zhang, Y. Gao, T. Darrell, K. Keutzer, and A. S. Vincentelli, "Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 834–13 844.

[29] B. Xu, Z. Zeng, C. Lian, and Z. Ding, "Few-shot domain adaptation via mixup optimal transport," *IEEE Transactions on Image Processing*, vol. 31, pp. 2518–2528, 2022.

[30] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," *Advances in neural information processing systems*, vol. 30, 2017.

[31] J. Zhang, Z. Chen, J. Huang, L. Lin, and D. Zhang, "Few-shot structured domain adaptation for virtual-to-real scene parsing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[32] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.

[33] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[34] N. Dong and E. P. Xing, "Few-shot semantic segmentation with prototype learning." in *BMVC*, vol. 3, no. 4, 2018.

[35] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9197–9206.

[36] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, 2020.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[39] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *European conference on computer vision*. Springer, 2008, pp. 30–43.

[40] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2239–2247.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[44] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*. Springer, 2016, pp. 102–118.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.
[46] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.
[47] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2020–2030.
[48] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3723–3732.
[49] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4500–4509.
[50] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*. PMLR, 2018, pp. 1989–1998.
[51] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2517–2526.
[52] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6758–6767.
[53] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured output via discriminative patch representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1456–1465.
[54] Y. Luo, P. Liu, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Category-level adversarial adaptation for semantic segmentation using purified features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[55] F. Lv, T. Liang, X. Chen, and G. Lin, "Cross-domain semantic segmentation via domain-invariant interactive relation transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4334–4343.
[56] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4085–4095.
[57] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 642–659.
[58] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang, "Label-driven reconstruction for domain adaptation in semantic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 480–498.
[59] L. Musto and A. Zinelli, "Semantically adaptive image-to-image translation for domain adaptation of semantic segmentation," *arXiv preprint arXiv:2009.01166*, 2020.
[60] B. Zhang, J. Xiao, and T. Qin, "Self-guided and cross-guided learning for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8312–8321.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

**Zhengeng Yang** received the B.S. and M.S. degrees from Central South University, Changsha, China, in 2009 and 2012, respectively. He received the Phd degree from Hunan University, Changsha, China, in 2020. He is currently a post-doctor researcher at Hunan University, Changsha. He was a Visiting Scholar with the University of Pittsburgh, Pittsburgh, PA during 2018 -2020. His research interests include computer vision, image analysis, and machine learning.

**Hongshan Yu** received the B.S., M.S. and Ph.D. degrees of Control Science and Technology from college of electrical and information engineering of Hunan University, Changsha, China, in 2001, 2004 and 2007 respectively. From 2011 to 2012, he worked as a postdoctoral researcher in Laboratory for Computational Neuroscience of University of Pittsburgh, USA. He joined the Hunan University as Assistant Professor in 2007 and became Professor in 2018. His research interests include autonomous robot and machine vision.

**Wei Sun** received the M.S. and Ph.D. degrees of Control Science and Technology from the Hunan University, Changsha, China, in 1999 and 2002, respectively. He is currently a Professor at Hunan University. His research interests include artificial intelligence, robot control, complex mechanical and electrical control systems, and automotive electronics.

**Li Cheng** received the PhD degree in computer science from the University of Alberta, Canada. He is a research scientist and group leader in Bioinformatics Institute (BII). Prior to joining BII July of 2010, he worked in Statistical Machine Learning group of NICTA, Australia, TTI-Chicago, and the University of Alberta, Canada. His research expertise is mainly on machine learning and computer vision. He is a senior member of the IEEE


**Ajmal Mian** is currently a Professor of computer science with The University of Western Australia. His research interests include computer vision, machine learning, 3D shape analysis, human action recognition, video description, and hyperspectral image analysis. He has received three prestigious fellowships from the Australian Research Council (ARC) and several awards including the West Australian Early Career Scientist of the Year Award, the Aspire Professional Development Award, the Vice-Chancellors Mid-Career Research Award, the Outstanding Young Investigator Award, the IAPR Best Scientific Paper Award, the EH Thompson Award, and excellence in Research Supervision Award. He has received several major research grants from the ARC, the National Health and Medical Research Council of Australia and the US Department of Defence. He serves as an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Image Processing, and the Pattern Recognition Journal.