Few-Shot Structured Domain Adaptation for Virtual-to-Real Scene Parsing

Junyi Zhang
Sun Yat-sen University
zhangjy329@mail2.sysu.edu.cn

Ziliang Chen
Sun Yat-sen University
c.ziliang@yahoo.com

Junying Huang
Sun Yat-sen University
huangjy229@mail2.sysu.edu.cn

Liang Lin
Sun Yat-sen University
linliang@ieee.org

Dongyu Zhang*
Sun Yat-sen University
zhangdy27@mail.sysu.edu.cn

Abstract

A structured domain adaptation (SDA) model for virtual-to-real scene parsing, learning to predict visual structure labels in real-world target scenes via mitigating the statistical discrepancy between large scale labeled virtual source and unlabeled real-world target images. But different from the source images drawn from urban simulation platforms, the target images could be expansive and difficult to collect at scale in real-world scenes. Besides, the trend of urbanization constantly changes the visual appearances of target scenes, which encourages SDA models to quickly adapt to new target scenes by merely given very few target images for training. To address the concerns, we attempt to achieve the virtual-to-real scene parsing from a new perspective inspired by few-shot learning. Instead of using a large amount of unlabeled target data used in existing SDA models, our few-shot SDA model takes a few of target real images with semantic labels in each scene, which collaborates with virtual source domain to train a virtual-to-real scene parser. Specifically, our framework is a two-stage adversarial network which contains a scene parser and two discriminators. Based on the data pairing method, our framework can handle the problem of scarce target data well and make full use of the limited semantic labels. We evaluate our method on two suites of virtual-to-real scene parsing setups. The experimental results show that our method exceeds the state-of-the-art SDA model by 7.1% in mIoU on SYNTHIA-to-CITYSCAPES and 4.03% in mIoU on GTA5-to-CITYSCAPES in the case of 1-shot.

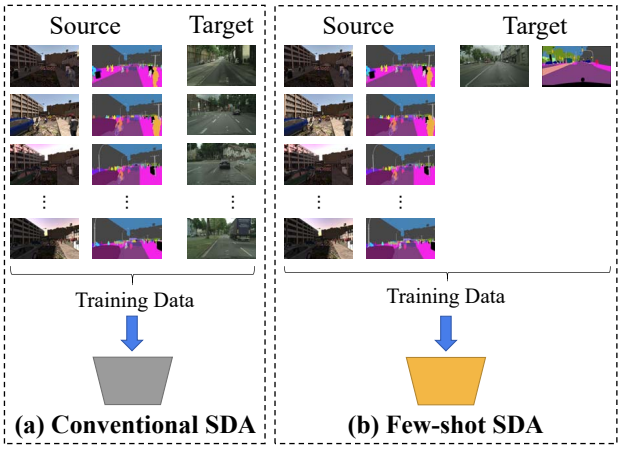

Figure 1: Comparison of conventional SDA and the proposed few-shot SDA. (a) is trained with supervised source data and a large amount of unsupervised target data. (b) is trained with supervised source data and very few supervised target data. Although conventional SDA can spare the cost of annotating a large amount of target data, but collecting them is also time-consuming and laborious. In contrast, the few-shot SDA only needs to collect a few target data and the cost of annotating them is completely acceptable.

1. Introduction

Resurrecting with huge-scale labeled databases [7], deep learning becomes the dominative technique to predict structured labels (*e.g.*, semantic masks) in diverse machine vision areas, *e.g.*, generic semantic segmentation [2, 1, 21, 38, 41], human body and scene parsing [12, 41, 39], *etc*. Among these visual structure prediction tasks, scene parsing attracts an increasing amount of attention due to its application potential in autonomous driving [26, 34]. However, building an urban scene large-scale labeled database exhausts labor efforts and can be quite expansive. To this

*Corresponding author is Dongyu Zhang. This work was supported in part by Natural Science Foundation of China under Grant No. 61876224, in part by the Natural Science Foundation of Guangdong Province under Grant No. 2017B010116001, and in part by the Fundamental Research Funds for the Central Universities.

end, plenty of current researches resort to virtual scene images, which can be handily generated by the computer-graphic programs within urban scene simulators [29, 28], with free machine-annotated semantic masks. Incorporating the synthetic labeled source images, structured domain adaptation (SDA) models [30, 35, 3] train the real-world scene parser by minimizing the discrepancy between the virtual-world and real-world domains [14]. The SDA models learn to transfer the virtual source semantic information into real-world target domain and therefore, which successfully spares the cost for annotating target scene images. Despite the impressive performances existing SDA models have already achieved, they may not truly overcome the application bottleneck in outdoor scene understanding. In particular, existing SDA strategies indeed save labor for manually annotating tremendous real-world scene images, whereas the vast expenses to create an urban scene benchmark, are not merely *due to manual annotation efforts*, but also *arise from the difficulty to collect available real-world scene images at scale*.

Unfortunately, the conventional SDA models [30, 35, 3] rely on a large number of target images to facilitate the label information transfer. Recent few-shot learning methods [9, 27, 31, 33, 20, 32] can categorize the new classes unseen in the training set, given only few examples of each new class. But they still require a lot of labeled data of old classes, and unable to use virtual data due to lack of domain adaptation ability. The related recent works are few-shot adversarial domain adaption (FADA) [25] and domain adaption in one-shot learning [8]. Both of them are applied to image classification task. Different from image classification, scene parsing may suffer from the complexity of high-dimensional features because most scenes, such as the autonomous driving scenes, are more complex, which makes the models used for image classification probably not suitable for outdoor scene parsing.

Attempting to address this concern, we propose a framework called *Few-shot Structured Domain Adaptation* (few-shot SDA). We find that the small amount of target data tends to make the training fluctuations larger when jointly training the model using target and source data, which is disadvantageous. Therefore, we propose a data pairing method for data enhancement to ensure stable training and reduce over-fitting. In addition, it is difficult to train the whole network effectively using very few target data, especially for the networks that are far away from the output. So we design our framework as a two-stage structure, the first-stage is a shallow auxiliary network and the second-stage is a deep network. The first-stage can not only enhance the adaptation of the low-level features but also provide an auxiliary prediction mask to guide the learning of the second-stage network. Benefiting from the auxiliary prediction mask, we propose a label filtering method, which is helpful to strengthen the learning of the network to the pixels which are difficult to classify. Finally, we employ spectral weight normalization [24] in the discriminators and propose a strategy of training alternately to further make the training more stable and effective. In general, the main contributions of this work are as follows:

- We provide a novel framework to handle the problem of few-shot structured domain adaptation for scene parsing. Our method can not only spare the cost for annotating target data but also greatly reduce the number of target data collected. To the best of our knowledge, we are the first to address this issue.
- Based on our proposed data pairing method, we design our framework as a two-stage structure with label filtering operation and provide an effective training strategy, which makes sense for solving the challenges in few-shot learning of scene parsing.
- Extensive experiments and evaluations on two suites of virtual-to-real scene parsing setups show that our proposed framework achieves superior performance in comparison to the state-of-the-art.

2. Related Work

Scene Parsing. Scene parsing is a fundamental topic in computer vision based on semantic segmentation. The goal is to map each pixel of an image into one of several predefined categories. With the development of deep learning, the pixel-level prediction tasks like scene parsing and semantic segmentation have achieved great progress. Fully convolutional network (FCN) [21] pioneered to replace fully-connected layers (FC) by convolutional layers, and many successive works [2, 40, 41] have further enhanced the accuracy and efficiency. There are also some works [35, 3, 30] that use the synthetic datasets based on rendering to handle the data annotation problem, as the labels are usually available directly from computers. We also use synthetic datasets, but it is necessary to narrow the domain shift between the synthetic data domain and real-world data domain.

Domain Adaptation. Domain adaptation technique can bridge the gap between the distribution of the source domain and the target domain. The earlier methods mainly involved using feature re-weighting techniques [6], or constructing intermediate representations using manifolds [13, 11]. Due to the power of deep neural networks (DNNs), the emphasis has shifted to aligning features extracted from the networks in an end-to-end manner. Adversarial learning [16, 10, 23] is one of the approaches. We focus on adversarial approaches because they are more relevant to our work and have achieved remarkable results in visual domain adaption. Ganin et al. [10] proposed the domain adversarial

neural network to transfer the feature distribution. Thereafter, many variants have been proposed with different loss functions [36, 22] or classifiers [23].

Few-Shot Learning. Few-shot learning aims to recognize novel visual categories from a limited amount of labeled training data. Recent few-shot learning works are mainly designed for image classification [33, 20, 32] and semantic segmentation [9, 27, 31], but they still require a lot of labeled data of old classes, and unable to use the virtual data directly due to the domain shift. The most relevant works are few-shot adversarial domain adaption (FADA) [25]. FADA used data pairing method for data enhancement, we are inspired by this idea.

3. Methodology

In this paper, we present our methodology for few-shot structured domain adaptation (SDA) for virtual-to-real scene parsing. Firstly, we describe the relevant definition of the problem. Secondly, we introduce the overall structure of our framework. Thirdly, we introduce the scene parser module, which includes the label filtering operation and the calculation of the segmentation loss. Finally, we introduce the balance domain pair adversarial module, which involves the process of adversarial learning. The complete pipeline is illustrated in Figure 2.

3.1. Problem Definition

We consider the task as few-shot pixel-level classifier learning. In the settings of our method, we only need to collect K images from each region, where K can be very small, or even 1. We are given two image pair sets $X_s = \{(x_n^s, y_n^s)\}_{n=1}^N$ and $X_t = \{(x_n^t, y_n^t)\}_{n=1}^M$, where X_s is source (synthetic) data, and X_t is a small part of target (real-world) data. Due to the memory limitation, we set the batch size to 1. In each iteration, we take two images x_n^s and x_n^t. We set $x \in \{x_n^s, x_n^t\}$. Specifically, each iteration training involves four images $\{x_n^t, x_n^s, x_{n-1}^t, x_{n-1}^s\}$, where x_n^t and x_n^s are the input images of current iteration. x_{n-1}^t and x_{n-1}^s are the input images of previous iteration. n represents the number of the iteration. In the target dataset that we select, the images in the training set are collected from N cities. We define *K*-shot as randomly selecting K images from each of the N cities. We consider (1-5)-shot (*K* = 1, 2, 3, 4, 5) settings.

3.2. Framework

Our proposed few-shot SDA framework is composed of a *Scene Parser Module* G and a *Balance Domain Pair Adversarial Module* D. We find that the low-level features which are far away from the output may not be adapted well. Therefore, we design our framework as a two-stage structure to enhance the adaptation of them. The first-stage (S_1) outputs the auxiliary mask (p_1). The second-stage (S_2) outputs the semantic mask (p_2). S_1 can guide the learning of S_2 by the label filtering operation using the auxiliary mask. D contains an auxiliary domain discriminator (D_1) and a domain discriminator (D_2). S_2 is the test pipeline, our goal is to learn a scene parser network G so that $S_2(v)$ can perform well in the test phase, where v represents the test image. It should be noted that both G and D will be used in the training phase. But during testing, only G will be used and D will be discarded.

Table 1: Characterizations of conventional SDA (CSDA) model and our proposed few-shot SDA (FSDA) model.

Method	Number of target image	Number of target label	Convergence
CSDA	2975	0	slow
FSDA	18 (1-shot)	18 (1-shot)	fast

3.3. Scene Parser Module

We use Resnet-101 [15] as base network of the scene parser. As shown in Figure 2, we calculate the first segmentation loss ($\mathcal{L}_{seg1}$) with the auxiliary mask and the original label. Then we filter the original label using the auxiliary mask and calculate the second segmentation loss ($\mathcal{L}_{seg2}$) with the semantic mask and the filtered label. We set the original label for the input image x as y.

Label Filtering. In order to make the second-stage have a learning focus, the original label is filtered according to the auxiliary mask, which can improve the classification accuracy of pixels that are difficult to identify. This label filtering operation can be expressed as:

$$\hat{y} = F(y, p_1, \beta) \tag{1}$$

where y is the original semantic label, $\hat{y}$ is the filtered label, β is a threshold and p_1 is the auxiliary mask. When the confidence of a pixel is higher than β, this means the pixel is relatively easy to recognize. So we set the category on the label of such easily identifiable pixels to *omitted category*, which means we do not calculate the segmentation loss of the *omitted category* in the second-stage. This allows the second-stage can focus more on learning difficult pixels, such as some edge pixels of objects.

Segmentation Loss. We adopt cross-entropy loss to calculate the segmentation loss. The sum of segmentation losses of S_1 and S_2 can be expressed as:

$$\begin{aligned}\mathcal{L}_{seg} &= \mathcal{L}_{seg1} + \lambda_{seg}\mathcal{L}_{seg2} \\ &= -\sum_{h,w}\sum_{c\in C}(y \log p_1 + \lambda_{seg}\hat{y} \log p_2)\end{aligned} \tag{2}$$

where λ_{seg} is the weight used to balance two segmentation losses, C is the number of categories and *w, h* is the width

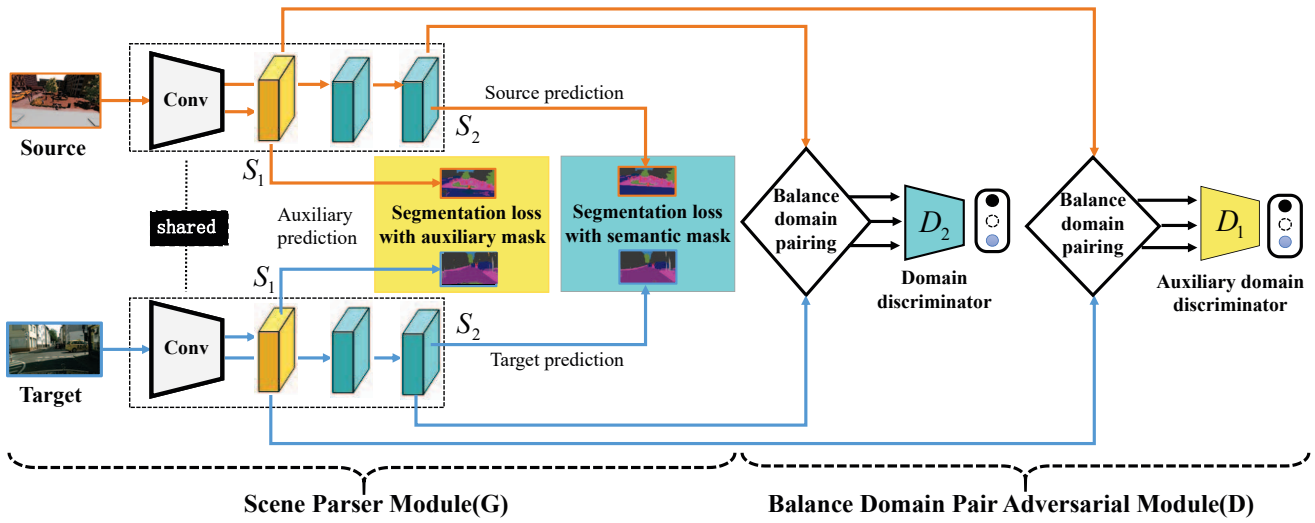

Figure 2: Framework overview. In each iteration, we pass a source image and a target image through the scene parser to obtain output predictions. The first-stage (S_1) outputs an auxiliary mask that is used to calculate the first segmentation loss ($\mathcal{L}_{seg1}$) and also can be used to filter the original label. The second-stage (S_2) outputs a semantic mask that is used to calculate the second segmentation loss ($\mathcal{L}_{seg2}$). D_1 and D_2 are used to distinguish the type of the input. The adversarial loss is calculated on the predictions of two discriminators and is back-propagated to the scene parser.

and height of the feature map, y is the original label, $\hat{y}$ is the filtered label, p_1 is the auxiliary mask and p_2 is the semantic mask.

3.4. Balance Domain Pair Adversarial Module

As shown in Figure 3, the inputs of *Balance domain pairing* module are two predictions of the current iteration and two predictions of the previous iteration. It should be noted that the first iteration does not have the previous iteration, so the input data will not be passed to *Balance Domain Pair Adversarial Module*. D_1 and D_2 no longer distinguish whether the input is from source domain or target domain, but the three types of pairs $\{g_{ss}, g_{st}, g_{tt}\}$ we defined. Next, We introduce how to adapt G via adversarial learning.

Training Generator. We use Euclidean distance to calculate the adversarial loss. The sum of adversarial losses can be expressed as:

$$\begin{aligned}\mathcal{L}_{adv} &= \lambda_{adv1}\mathcal{L}_{adv1} + \lambda_{adv2}\mathcal{L}_{adv2} \\ &= \lambda_{adv1}\sum_{h,w}\sum_{j\in O}\|D_1(g_j) - z_{ss}\|^2 \\ &+ \lambda_{adv2}\sum_{h,w}\sum_{j\in O}\|D_2(g_j) - z_{ss}\|^2\end{aligned} \tag{3}$$

where λ_{adv1} and λ_{adv2} are the weights used to balance the adversarial losses, z_{ss} is the label corresponding to g_{ss}. $O = \{st, tt\}$.

Training Discriminator. We train D_1 and D_2 separately and still use Euclidean distance to calculate the loss. For

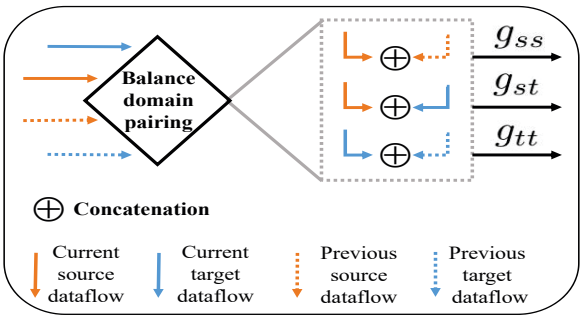

Figure 3: This is the details of the *Balance domain pairing* module in Figure 2. The solid line represents the data flow of the current iteration. The dotted line represents the data flow of the previous iteration. g_{tt} is a pair of two target images. g_{ss} is a pair of two source images. and g_{st} is a pair of one target image and one source image.

D_1, the loss can be written as:

$$\mathcal{L}_d = \sum_{h,w}\sum_{j\in Q}\|D_1(g_j) - z_j\|^2 \tag{4}$$

where z_j is the label corresponding to g_j, Q=$\{ss, st, tt\}$. For D_2, the loss is calculated in the same way as D_1.

The training objective for scene parser module can be written as:

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \mathcal{L}_{adv} \tag{5}$$

We can optimize the following min-max criterion:

$$\max_{D} \min_{G} \mathcal{L}_{total} \tag{6}$$

4. Implementation Details

Scene Parser. We adopt the DeepLab-v2 framework with ResNet-101 [15] pre-trained on ImageNet by replacing the final classifier with the Atrous Spatial Pyramid Pooling module (ASPP) [2] as our base network. We set the stride of the last two convolutional layers from 2 to 1, so that the resolution of the output features can be effectively mapped to 1/8 of the input image size. Finally, we add an up-sampling layer along with the softmax output to match the size of the input images.

Discriminator. For D_1 and D_2, the structure is the same, but they are two independent discriminators. In order to preserve the spatial information, we utilize 5 all full-convolutional layers with kernel 4×4, stride of 2, padding of 1 in D_1 and D_2. The channel number of the layers is $\{64, 128, 256, 512, 1\}$. Except that the last layer, which is followed a *tanh* to limit the output value between -1 and 1, the other four layers are followed by a *leaky ReLU*. The final outputs of the discriminators are feature maps. Due to the memory limitation, we train the discriminators with scene parser using a small batch size, so do not use any batch-normalization layers [18]. To stabilize the training of the discriminators, we use spectral normalization [24] after each convolutional layer.

We implement our framework using PyTorch toolbox on two GTX 1080Ti GPUs with 12GB memory. To train the scene parser, we use the stochastic gradient descent (SGD) with Nesterov acceleration where the momentum is 0.9 and the weight decay is 10^{-4}. The initial learning rate is set as 2.5×10^{-4}. To train the discriminators, we adopt two Adam optimizers [19] where the initial learning rate is 10^{-4} and the momentum is 0.9 and 0.99. The learning rates of all optimizers are decreased using the polynomial decay with a power of 0.9 as mentioned in [2]. The β is set as 0.95 in Equation (1). Finally, we set the λ_{seg} to 0.1 in Equation (2) and set the λ_{adv1} to 0.0002, λ_{adv2} to 0.001 in Equation (3).

5. Experiments and Results

In this section, we provide a quantitative evaluation of our proposed method by carrying out the experiments on SYNTHIA-to-CITYSCAPES and GTA5-to-CITYSCAPES. Our entire training strategy contains two phases: the first is adversarial learning, i.e, training the scene parser and discriminators jointly, the second is training the scene parser independently using target data. We gradually reduce the domain shift by performing the first training phase and the second training phase alternately in an end-to-end manner.

5.1. Experiment Setup

We set SYNTHIA [29] and GTA5 [28] as source domain and set CITYSCAPES [5] as target domain. In all experiments, we use the IoU metric. Next, we briefly introduce the datasets related to our experiments below:

CITYSCAPES is a real-world image dataset which consists of 2975 images in the training set, and 500 images in the verification set. There are 18 sub-folders in the training set representing 18 different cities. The resolution of the images is 2048×1024. And the pixel-level labels of 19 semantic categories are provided. Our few-shot target domain images are randomly selected from the training set. Finally, we use the verification set to test the trained model.

SYNTHIA is a synthetic dataset of urban scenes, it contains 9400 images compatible with the CITYSCAPES annotated categories. Similar to [35, 4], in all the experiments with SYNTHIA as the source domain, we evaluate on the CITYSCAPES verification set with 13 categories.

GTA5 is a synthetic dataset which contains 24966 images with the resolution of 1914×1052. The images are from a video game based on the city of Los Angeles. There are 19 semantic categories compatible with CITYSCAPES dataset. In all the experiments with GTA5 as the source domain, we evaluate on the CITYSCAPES verification set with 19 categories.

5.2. SYNTHIA → CITYSCAPES

Table 2 shows the comparison results of our approach with the methods of conventional SDA, joint training and fine-tuning. Due to the different experimental settings and training strategy, the results of source-only model and fully-supervised (Oracle) model reported by us do not match with the results reported in [35].

Our method achieves a mIoU of 53.8 in the 1-shot experiment, achieving 7.1% improvement over the best performing conventional SDA method [35]. This shows that our approach has a significant performance advantage over the conventional SDA methods. It also can be seen from the comparison results that our method exceeds fine-tuning by 4.45% (1-shot), 4.84% (2-shot), 3.36% (3-shot), 3.83% (4-shot) and 4.75% (5-shot). In addition, the performance of joint training is similar to that of fine-tuning. Therefore, we think that fine-tuning has little effect on domain adaptation in the case of few-shot, which illustrates the necessity of our method. Obviously, sufficient experiments demonstrates the effectiveness and stability of our method. Figure 4 shows the comparison of visualization results, indicating that our method has significant improvements in recognizing street lights, signal lights, persons and bicycles, etc.

5.3. GTA5 → CITYSCAPES.

Similar to the previous experiments. Table 3 reports the performance of our proposed method in comparison with

Table 2: Mean IoU Results of adapting SYNTHIA-to-CITYSCAPES. We compare our results with the conventional SDA, FT, and JT. FT denotes fine-tuning. JT denotes jointly training the scene parser using supervised source data and few supervised target data.

		Synthia → *Cityscapes*													
Base	Method	road	sidewalk	building	light	sign	vegetation	sky	person	rider	car	bus	motocycle	bicycle	mIoU
VGG-16	FCN wild [17]	11.5	19.6	30.8	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	22.9
	CDA [37]	65.2	26.1	74.9	3.5	3.0	76.1	70.6	47.1	8.2	43.2	20.7	0.7	13.1	34.8
	Cross-City [4]	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
	MAA (single-level) [35]	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
Resnet-101	Source-only (Ours)	60.12	22.38	66.56	4.63	7.15	75.22	76.58	33.35	10.55	54.53	5.63	1.25	17.65	33.51
	MAA [35]	84.3	42.7	77.5	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7
Resnet-101	JT (1-shot)	91.24	53.18	79.1	14.81	30.97	82.61	81.8	49.81	16.08	78.67	15.93	5.23	42.47	49.38
	FT (1-shot)	91.53	53.1	80.04	12.84	26.78	83.49	81.9	52.8	9.33	78.36	21.18	3.44	46.82	49.35
	Ours (1-shot)	**92.74**	**56.54**	**82.06**	**17.41**	**32.95**	**84.95**	**84.55**	**56.12**	**23.86**	**82.02**	**25.59**	**8.36**	**52.27**	**53.8 (+4.45/+4.42)**
Resnet-101	JT (2-shot)	92.03	54.31	80.69	14.19	27.68	82.52	**83.18**	50.94	**15.98**	78.3	16.93	7.54	46.27	50.04
	FT (2-shot)	90.69	51.36	80.43	16.85	25.71	83.55	78.71	54.56	12.09	**79.98**	15.05	6.4	47.42	49.45
	Ours (2-shot)	**92.68**	**55.44**	**82.25**	**24.54**	**35.43**	**85.08**	83.09	**57.01**	14.95	79.87	**24.47**	**17.91**	**53.02**	**54.29 (+4.84/+4.25)**
Resnet-101	JT (3-shot)	93.43	58.56	82.33	**29.98**	35.64	83.75	86.72	52.71	19.3	81.62	7.05	6.36	51.49	53.0
	FT (3-shot)	92.92	57.3	82.95	19.04	30.76	84.47	86.49	57.16	27.9	80.52	8.5	13.39	54.12	53.5
	Ours (3-shot)	**93.96**	**60.97**	**83.84**	27.95	**40.46**	**85.83**	**88.27**	**59.44**	**28.58**	**84.27**	**11.6**	**15.38**	**58.59**	**56.86 (+3.36/3.86)**
Resnet-101	JT (4-shot)	**94.7**	**64.52**	82.88	28.11	38.33	84.5	87.0	55.88	31.91	81.97	6.73	2.7	52.69	54.76
	FT(4-shot)	93.91	61.47	82.69	20.91	32.75	84.44	83.27	56.85	31.61	81.45	**13.06**	6.06	54.81	54.1
	Ours (4-shot)	94.36	62.68	**84.02**	**29.69**	**42.01**	**85.94**	**87.72**	**59.16**	**35.83**	**83.12**	10.58	**20.02**	**58.0**	**57.93 (+3.83/3.17)**
Resnet-101	JT (5-shot)	94.17	61.6	82.98	27.27	41.55	84.32	85.65	54.81	26.68	82.57	28.56	15.55	50.5	56.63
	FT (5-shot	93.48	58.73	83.0	20.64	37.22	85.25	81.84	57.58	**30.77**	81.8	25.31	22.61	52.22	56.19
	Ours (5-shot)	**94.41**	**63.31**	**84.47**	**30.91**	**50.87**	**86.05**	**88.19**	**61.34**	28.17	**86.32**	**35.5**	**24.17**	**58.56**	**60.94 (+4.75/+4.31)**

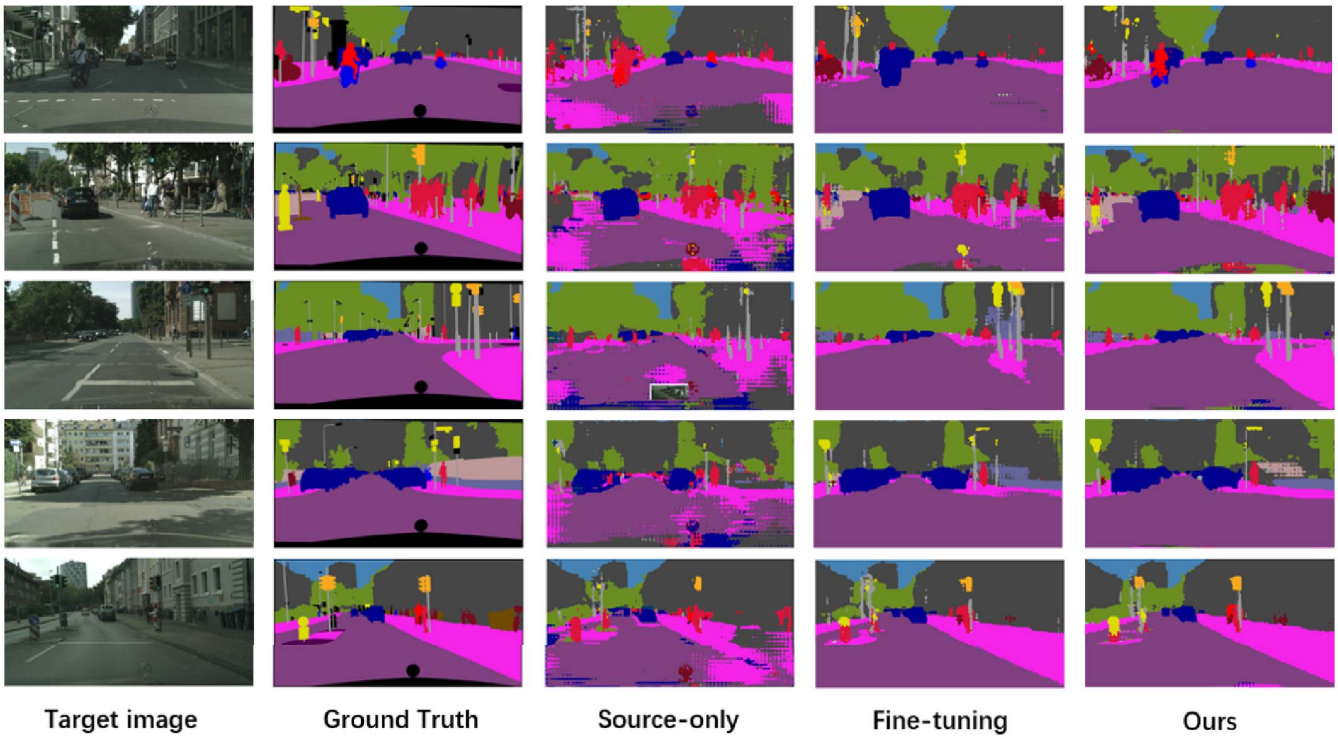

Figure 4: Example results of adapted scene parsing on SYNTHIA-to-CITYSCAPES. For each test image, we show the source-only (before adaptation), fine-tuning and our adapted results in the output space.

Table 3: Mean IoU results of adapting GTA5-to-CITYSCAPES. We compare our results with the conventional SDA, FT, and JT. FT denotes fine-tuning. JT denotes jointly training the scene parser using supervised source data and few supervised target data.

$Gta5 \rightarrow Cityscapes$																					
Base	Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	trunk	bus	train	motor	bike	mIoU
VGG-16	FCN wild [17]	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
	CyCADA (pixel) [16]	83.5	38.3	76.4	20.6	16.5	22.2	26.2	21.9	80.4	28.7	65.7	49.4	4.2	74.6	16.0	26.6	2.0	8.0	0.0	34.8
	MAA (single) [35]	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
	LSD [30]	88.0	30.5	78.6	25.2	23.5	16.7	23.5	11.6	78.7	27.2	71.9	51.3	19.5	80.4	19.8	18.3	0.9	20.8	18.4	37.1
Resnet-101	Source-only (Ours)	85.94	40.34	81.4	24.19	16.63	26.58	28.3	15.04	79.75	27.5	83.47	49.81	20.91	71.97	22.11	21.03	0.01	18.41	24.05	38.76
	ROAD [3]	76.3	36.1	69.6	28.8	22.4	28.6	29.3	14.8	82.3	35.3	72.9	54.4	17.8	78.9	27.7	30.3	4.0	24.9	12.6	39.4
	MAA [35]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
Resnet-101	JT (1-shot)	90.79	56.41	81.57	**33.85**	18.45	30.64	**29.31**	28.73	83.66	37.27	**84.71**	49.69	17.52	69.5	24.68	27.1	**1.98**	16.47	26.15	42.55
	FT (1-shot)	93.41	60.58	82.72	21.19	**23.93**	30.66	27.04	29.57	84.45	39.13	66.37	52.32	20.44	84.54	**35.7**	**29.65**	1.05	12.81	33.76	43.65
	Ours (1-shot)	**94.49**	**62.67**	**82.76**	25.03	19.23	**32.59**	28.83	**36.9**	**84.71**	**39.81**	83.55	**54.87**	**25.19**	**84.87**	31.85	29.39	0.0	**16.8**	**48.56**	**46.43 (+2.78/+3.88)**
Resnet-101	JT (2-shot)	92.03	53.0	82.93	32.69	24.66	32.06	30.46	30.75	84.45	41.44	84.59	52.4	13.22	80.98	**42.19**	**37.24**	1.43	11.26	42.34	45.8
	FT (2-shot)	**94.51**	**62.45**	83.62	23.88	**28.17**	32.05	31.72	30.47	84.84	39.97	78.62	53.14	8.76	**85.09**	34.58	28.3	**16.66**	14.19	40.76	45.88
	Ours (2-shot)	94.07	61.62	**84.68**	**35.9**	25.56	**34.29**	**34.51**	**37.83**	**86.41**	**43.65**	**85.07**	**55.97**	**17.91**	84.54	39.74	36.03	2.22	**21.94**	**51.37**	**49.12 (+3.24/+3.32)**
Resnet-101	JT (3-shot)	92.43	54.1	83.0	33.81	24.46	32.49	34.29	35.76	84.91	39.82	86.13	53.72	25.5	82.27	31.04	**30.89**	12.38	19.52	45.38	47.47
	FT (3-shot)	**94.65**	63.45	84.72	26.2	25.69	34.64	34.45	38.06	**86.29**	41.93	**88.55**	56.28	24.52	**86.04**	**32.1**	7.52	**20.7**	26.42	44.78	48.26
	Ours (3-shot)	94.34	**64.64**	**85.25**	**35.63**	**27.35**	**36.13**	**36.93**	**40.12**	86.25	**45.17**	85.85	**58.3**	**31.09**	83.39	31.82	29.28	11.25	**29.2**	**56.42**	**50.97 (+2.71/+3.5)**
Resnet-101	JT (4-shot)	92.67	54.58	83.27	29.0	25.25	34.16	32.36	34.87	84.6	41.82	82.01	54.71	26.97	80.7	36.8	28.49	**3.16**	20.33	48.1	47.04
	FT (4-shot)	94.35	**65.17**	84.54	28.67	**30.52**	34.82	32.99	39.72	85.78	44.85	77.77	56.45	25.08	**86.05**	13.17	**31.72**	0.03	7.76	45.3	46.57
	Ours (4-shot)	**94.42**	62.99	**85.21**	**38.52**	29.34	**35.36**	**33.16**	**45.07**	**86.73**	**45.11**	**88.83**	**59.18**	**32.64**	85.6	**40.65**	29.17	0.0	**21.08**	**53.64**	**50.88 (+4.31/+3.84)**
Resnet-101	JT (5-shot)	92.78	59.9	83.72	31.11	25.92	32.14	34.69	42.54	84.63	40.51	84.36	54.16	26.09	79.25	39.23	**43.38**	**2.71**	8.86	49.34	48.18
	FT (5-shot)	**95.26**	**65.53**	84.91	22.5	**28.87**	34.91	33.38	42.07	86.36	42.83	83.86	55.33	27.08	**86.43**	**47.8**	43.64	3.91	23.08	44.14	50.1
	Ours (5-shot)	94.61	65.12	**85.57**	**33.68**	27.25	**37.31**	**36.75**	**48.63**	**86.79**	**47.94**	**87.56**	**60.51**	**32.12**	85.68	41.18	**45.27**	**14.75**	**32.1**	**54.6**	**53.55 (+3.45/+5.37)**

Table 4: Ablation study shows the effect of each component on the final performance of our method on SYNTHIA-to-CITYSCAPES.

Method	mean IoU
Source-only	33.51
Ours w/o data paring	48.35
Ours w/o two-stage	51.31
Ours w/o training alternately	51.75
Ours w/o filtering	52.13
Ours (1-shot)	**53.8**

the conventional SDA, joint training and fine-tuning. The mean IOU of our method is 46.43 in the case of 1-shot, which is 4.03% higher than the existing best-performing conventional SDA method [35]. Besides, our method exceeds fine-tuning by 2.78% (1-shot), 3.24% (2-shot), 2.71% (3-shot), 4.31% (4-shot) and 3.45% (5-shot). The performance of the joint training is also similar to that of the fine-tuning.

5.4. Analysis

Ablation Study. In this experiment, we show how each component in our framework affects the final performance. We consider 5 cases: (a) Ours w/o data paring: we do not pair source data with target data. In this case, the discriminator distinguishes whether the input is from the source domain or the target domain. (b) Ours w/o two-stage: the first-stage is discarded, we only use the second-stage. (c) Ours w/o training alternately: we only perform the first training phase, i.e, jointly training the scene parser and discriminators. (d) Ours w/o filtering: the semantic label of the input image is not filtered and is used directly by the second-stage. (e) Ours: the full implementation of our method. The mean IoU results are reported in Table 4. It can be observed that each component in our framework is of great importance to obtain full improvement in test performance.

Analysis of Convergence Performance. We compare the convergence performance on segmentation loss of our method (1-shot) with the conventional SAD method [35]. Figure 5 shows that our method converges faster and more smoothly than the conventional SAD method [35]. As can be seen from the graph, our model converges after 10000 iterations of training, and the loss fluctuation is relatively stable. While the conventional SAD model still does not converge after 20000 iterations of training. We think this is because our model uses far fewer target data, so it converges faster and the training is more stable under the supervision of a few target semantic labels.

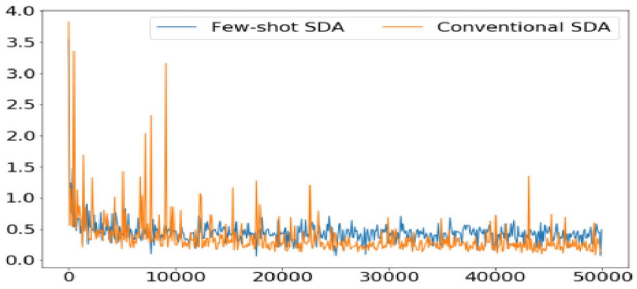

Figure 5: The convergence performance on segmentation loss of our few-shot SDA model and the conventional SAD model [35]. The horizontal axis represents the number of iterations. Best viewed in color.

6. Conclusion

In this paper, we presented a novel framework to address the problem of few-shot structured domain adaptation for virtual-to-real scene parsing. Even if only one target image is available for each scene, our framework can work well. The experimental results on GTA5-to-CITYSCAPES and SYNTHIA-to-CITYSCAPES have demonstrated the effectiveness of our method. We also have set a new state-of-the-art in the experiments.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[3] Y. Chen, W. Li, and L. Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018.

[4] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017.

[5] 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*, pages 3213–3223, 2016.

[6] H. Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[8] N. Dong and E. P. Xing. Domain adaption in one-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2018.

[9] N. Dong and E. P. Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, page 4, 2018.

[10] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[11] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[12] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin. Instance-level human parsing via part grouping network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–785, 2018.

[13] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[14] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[15] 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*, pages 770–778, 2016.

[16] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[17] 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.

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014.

[20] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[22] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[23] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[24] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[25] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6670–6680, 2017.

[26] X. Pan, Y. You, Z. Wang, and C. Lu. Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952*, 2017.

[27] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine. Few-shot segmentation propagation with guided networks. *arXiv preprint arXiv:1806.07373*, 2018.

[28] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016.

[29] 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*, pages 3234–3243, 2016.

[30] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. *arXiv preprint arXiv:1711.06969*, 2017.

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

[32] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[33] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[34] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018.

[35] 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. *arXiv preprint arXiv:1802.10349*, 2018.

[36] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[37] Z. Yang, P. David, and B. Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. 2017.

[38] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[39] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Perspective-adaptive convolutions for scene parsing. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[40] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[41] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.