Learning to Adapt Structured Output Space for Semantic Segmentation

Yi-Hsuan Tsai[1]* Wei-Chih Hung[2]* Samuel Schulter[1] Kihyuk Sohn[1]
Ming-Hsuan Yang[2] Manmohan Chandraker[1]
[1]NEC Laboratories America [2]University of California, Merced

Abstract

Convolutional neural network-based approaches for semantic segmentation rely on supervision with pixel-level ground truth, but may not generalize well to unseen image domains. As the labeling process is tedious and labor intensive, developing algorithms that can adapt source ground truth labels to the target domain is of great interest. In this paper, we propose an adversarial learning method for domain adaptation in the context of semantic segmentation. Considering semantic segmentations as structured outputs that contain spatial similarities between the source and target domains, we adopt adversarial learning in the output space. To further enhance the adapted model, we construct a multi-level adversarial network to effectively perform output space domain adaptation at different feature levels. Extensive experiments and ablation study are conducted under various domain adaptation settings, including synthetic-to-real and cross-city scenarios. We show that the proposed method performs favorably against the state-of-the-art methods in terms of accuracy and visual quality.

1. Introduction

Semantic segmentation aims to assign each pixel a semantic label, e.g., person, car, road or tree, in an image. Recently, methods based on convolutional neural networks (CNNs) have achieved significant progress in semantic segmentation [2, 21, 23, 24, 40, 42, 43] with applications for autonomous driving [9] and image editing [36]. The crux of CNN-based approaches is to annotate a large number of images that cover possible scene variations. However, this trained model may not generalize well to unseen images, especially when there is a domain gap between the training (source) and test (target) images. For instance, the distribution of appearance for objects and scenes may vary in different cities, and even weather and lighting conditions can change significantly in the same city. In such cases, relying only on the supervised model that requires re-annotating per-pixel ground truths in different scenarios would entail prohibitively high labor cost.

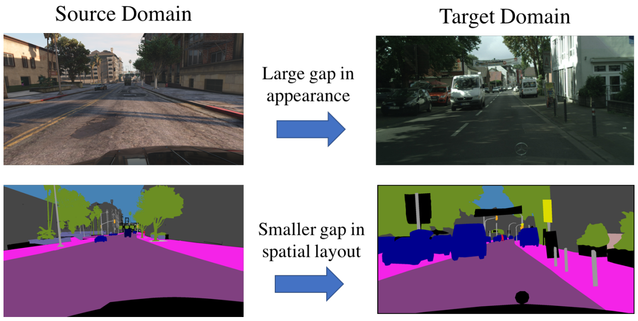

Figure 1. Our motivation of learning adaptation in the output space. While images may be very different in appearance, their outputs are structured and share many similarities, such as spatial layout and local context.

To address this issue, knowledge transfer or domain adaptation techniques have been proposed to close the gap between source and target domains, where annotations are not available in the target domain. For image classification, one effective approach is to align features across two domains [8, 25] such that the adapted features can generalize to both domains. Similar efforts have been made for semantic segmentation via adversarial learning in the feature space [3, 13]. However, different from the image classification task, feature adaptation for semantic segmentation may suffer from the complexity of high-dimensional features that needs to encode diverse visual cues, including appearance, shape and context. This motivates us to develop an effective method for adapting pixel-level prediction tasks rather than using feature adaptation. In semantic segmentation, we note that the output space contains rich information, both spatially and locally. For instance, even if images from two domains are very different in appearance, their segmentation outputs share a significant amount of similarities, e.g., spatial layout and local context (see Figure 1). Based on this observation, we address the pixel-level domain adaptation problem in the output (segmentation) space.

[*] Both authors contribute equally to this work.

In this paper, we propose an end-to-end CNN-based domain adaptation algorithm for semantic segmentation. Our formulation is based on adversarial learning in the output space, where the intuition is to directly make the predicted label distributions close to each other across source and target domains. Based on the generative adversarial network (GAN) [10, 31, 22], the proposed model consists of two parts: 1) a segmentation model to predict output results, and 2) a discriminator to distinguish whether the input is from the source or target segmentation output. With an adversarial loss, the proposed segmentation model aims to fool the discriminator, with the goal of generating similar distributions in the output space for either source or target images.

The proposed method also adapts features as the errors are back-propagated to the feature level from the output labels. However, one concern is that lower-level features may not be adapted well as they are far away from the high-level output labels. To address this issue, we develop a multi-level strategy by incorporating adversarial learning at different feature levels of the segmentation model. For instance, we can use both *conv5* and *conv4* features to predict segmentation results in the output space. Then two discriminators can be connected to each of the predicted output for multi-level adversarial learning. We perform one-stage end-to-end training for the segmentation model and discriminators jointly, without using any prior knowledge of the data in the target domain. In the testing phase, we can simply discard discriminators and use the adapted segmentation model on target images, with no extra computational requirements.

Due to the high labor cost of annotating segmentation ground truth, there has been great interest in large-scale synthetic datasets with annotations, e.g., GTA5 [32] and SYNTHIA [33]. As a result, one critical setting is to adapt the model trained on synthetic data to real-world datasets, such as Cityscapes [4]. We follow this setting and conduct extensive experiments to validate the proposed domain adaptation method. First, we use a strong baseline model that is able to generalize to different domains. We note that a strong baseline facilitates real-world applications and can evaluate the limitation of the proposed adaptation approach. Based on this baseline model, we show comparisons using adversarial adaptation in the feature and output spaces. Furthermore, we show that the multi-level adversarial learning improves the results over single-level adaptation. In addition to the synthetic-to-real setting, we show experimental results on the Cross-City dataset [3], where annotations are provided in one city (source), while testing the model on another unseen city (target). Overall, our method performs favorably against state-of-the-art algorithms on numerous benchmark datasets under different settings.

The contributions of this work are as follows. First, we propose a domain adaptation method for pixel-level semantic segmentation via adversarial learning. Second, we demonstrate that adaptation in the output (segmentation) space can effectively align scene layout and local context between source and target images. Third, a multi-level adversarial learning scheme is developed to adapt features at different levels of the segmentation model, which leads to improved performance.

2. Related Work

Semantic Segmentation. State-of-the-art semantic segmentation methods are mainly based on the recent advances of deep neural networks. As proposed by Long *et al.* [24], one can transform a classification CNN (e.g., AlexNet [19], VGG [34], or ResNet [11]) to a fully-convolutional network (FCN) for semantic segmentation. Numerous methods have since been developed to improve this model by utilizing context information [15, 42] or enlarging receptive fields [2, 40]. To train these advanced networks, a substantial amount of dense pixel annotations must be collected in order to match the model capacity of deep CNNs. As a result, weakly and semi-supervised approaches [5, 14, 17, 29, 30] are proposed in recent years to reduce the heavy labeling cost of collecting segmentation ground truths. However, in most real-world applications, it is difficult to obtain weak annotations and the trained model may not generalize well to unseen image domains.

Another approach to tackle the annotation problem is to construct synthetic datasets based on rendering, e.g., GTA5 [32] and SYNTHIA [33]. While the data collection is less costly since the pixel-level annotation can be done with a partially automated process, these datasets are usually used in conjunction with real-world datasets for joint learning to improve the performance. However, when training solely on the synthetic dataset, the model does not generalize well to real-world data, mainly due to the large domain shift between synthetic images and real-world images, i.e., appearance differences are still significant with current rendering techniques. Although synthesizing more realistic images can decrease the domain shift, it is necessary to use domain adaptation to narrow the performance gap.

Domain Adaptation. Domain adaptation methods for image classification have been developed to address the domain-shift problem between the source and target domains. Numerous methods [7, 8, 25, 26, 35, 37, 38] are developed based on CNN classifiers due to performance gain. The main insight behind these approaches is to tackle the problem by aligning the feature distribution between source and target images. Ganin *et al.* [7, 8] propose the Domain-Adversarial Neural Network (DANN) to transfer the feature distribution. A number of variants have since been proposed with different loss functions [25, 37, 38] or classifiers [26]. Recently, the PixelDA method [1] addresses domain adaptation for image classification by transferring the source im-

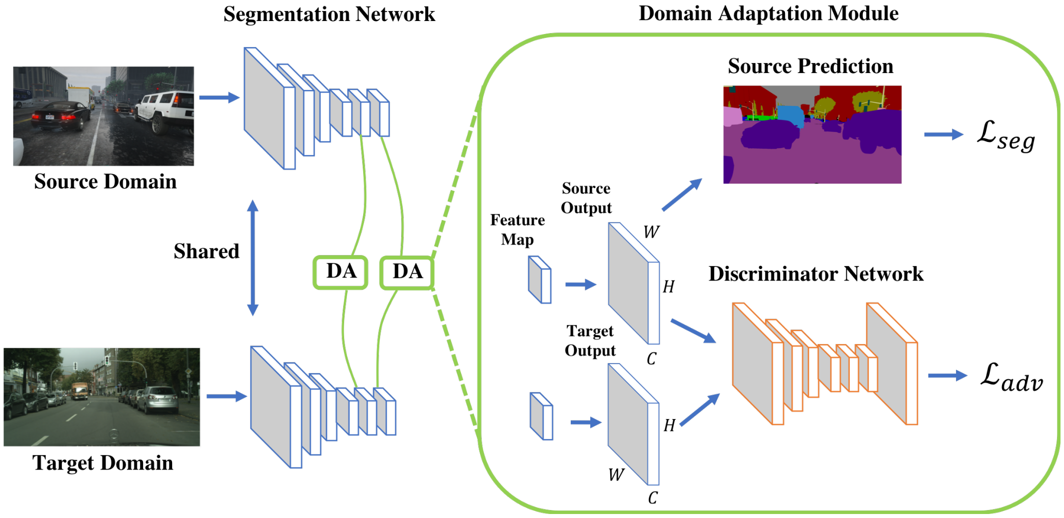

Figure 2. Algorithmic overview. Given images with the size H by W in source and target domains, we pass them through the segmentation network to obtain output predictions. For source predictions with C categories, a segmentation loss is computed based on the source ground truth. To make target predictions closer to the source ones, we utilize a discriminator to distinguish whether the input is from the source or target domain. Then an adversarial loss is calculated on the target prediction and is back-propagated to the segmentation network. We call this process as one adaptation module, and we illustrate our proposed multi-level adversarial learning by adopting two adaptation modules at two different levels here.

ages to target domain, thereby obtaining a simulated training set for target images.

We note that domain adaptation for pixel-level prediction tasks have not been explored widely. Hoffman *et al.* [13] introduce the task of domain adaptation on semantic segmentation by applying adversarial learning (i.e., DANN) in a fully-convolutional way on feature representations and additional category constraints similar to the constrained CNN [30]. Other methods focus on adapting synthetic-to-real or cross-city images by adopting class-wise adversarial learning [3] or label transfer [3]. Similar to the PixelDA method [1], one concurrent work, CyCADA [12] uses the CycleGAN [44] and transfers source domain images to the target domain with pixel alignment, thus generating extra training data combined with feature space adversarial learning [13].

Although feature space adaptation has been successfully applied to image classification, pixel-level tasks such as semantic segmentation remains challenging based on feature adaptation-based approaches. In this paper, we use the property that pixel-level predictions are structured outputs that contain information spatially and locally, to propose an efficient domain adaptation algorithm through adversarial learning in the output space.

3. Algorithmic Overview

3.1. Overview of the Proposed Model

Our domain adaptation algorithm consists of two modules: a segmentation network $\mathbf{G}$ and the discriminator $\mathbf{D}_i$, where i indicates the level of a discriminator in the multi-level adversarial learning. Two sets of images $\in \mathbb{R}^{H\times W\times 3}$ from source and target domains are denoted as $\{\mathcal{I}_S\}$ and $\{\mathcal{I}_T\}$. We first forward the source image I_s (with annotations) to the segmentation network for optimizing $\mathbf{G}$. Then we predict the segmentation softmax output P_t for the target image I_t (without annotations). Since our goal is to make segmentation predictions P of source and target images (i.e., P_s and P_t) close to each other, we use these two predictions as the input to the discriminator $\mathbf{D}_i$ to distinguish whether the input is from the source or target domain. With an adversarial loss on the target prediction, the network propagates gradients from $\mathbf{D}_i$ to $\mathbf{G}$, which would encourage $\mathbf{G}$ to generate similar segmentation distributions in the target domain to the source prediction. Figure 2 shows the overview of the proposed algorithm.

3.2. Objective Function for Domain Adaptation

With the proposed network, we formulate the adaptation task containing two loss functions from both modules:

$$\mathcal{L}(I_s, I_t) = \mathcal{L}_{seg}(I_s) + \lambda_{adv}\mathcal{L}_{adv}(I_t), \qquad (1)$$

where $\mathcal{L}_{seg}$ is the cross-entropy loss using ground truth annotations in the source domain, and $\mathcal{L}_{adv}$ is the adversarial loss that adapts predicted segmentations of target images to the distribution of source predictions (see Section 4). In (1), λ_{adv} is the weight used to balance the two losses.

4. Output Space Adaptation

Different from image classification based on features [8, 25] that describe the global visual information of the image, high-dimensional features learned for semantic segmentation encodes complex representations. As a result, adaptation in the feature space may not be the best choice for semantic segmentation. On the other hand, although segmentation outputs are in the low-dimensional space, they contain rich information, e.g., scene layout and context. Our intuition is that no matter images are from the source or target domain, their segmentations should share strong similarities, spatially and locally. Thus, we utilize this property to adapt low-dimensional softmax outputs of segmentation predictions via an adversarial learning scheme.

4.1. Single-level Adversarial Learning

Discriminator Training. Before introducing how to adapt the segmentation network via adversarial learning, we first describe the training objective for the discriminator. Given the segmentation softmax output $P = \mathbf{G}(I) \in \mathbb{R}^{H \times W \times C}$, where C is the number of categories, we forward P to a fully-convolutional discriminator $\mathbf{D}$ using a cross-entropy loss $\mathcal{L}_d$ for the two classes (i.e., source and target). The loss can be written as:

$$\begin{aligned}\mathcal{L}_d(P) = -\sum_{h,w} & (1-z)\log(\mathbf{D}(P)^{(h,w,0)}) \\ & + z\log(\mathbf{D}(P)^{(h,w,1)}),\end{aligned} \tag{2}$$

where $z = 0$ if the sample is drawn from the target domain, and $z = 1$ for the sample from the source domain.

Segmentation Network Training. First, we define the segmentation loss in (1) as the cross-entropy loss for images from the source domain:

$$\mathcal{L}_{seg}(I_s) = -\sum_{h,w}\sum_{c \in C} Y_s^{(h,w,c)} \log(P_s^{(h,w,c)}), \tag{3}$$

where Y_s is the ground truth annotations for source images and $P_s = \mathbf{G}(I_s)$ is the segmentation output.

Second, for images in the target domain, we forward them to $\mathbf{G}$ and obtain the prediction $P_t = \mathbf{G}(I_t)$. To make the distribution of P_t closer to P_s, we use an adversarial loss $\mathcal{L}_{adv}$ in (1) as:

$$\mathcal{L}_{adv}(I_t) = -\sum_{h,w} \log(\mathbf{D}(P_t)^{(h,w,1)}). \tag{4}$$

This loss is designed to train the segmentation network and fool the discriminator by maximizing the probability of the target prediction being considered as the source prediction.

4.2. Multi-level Adversarial Learning

Although performing adversarial learning in the output space directly adapts predictions, low-level features may not be adapted well as they are far away from the output. Similar to the deep supervision method [20] that uses auxiliary loss for semantic segmentation [42], we incorporate additional adversarial module in the low-level feature space to enhance the adaptation. The training objective for the segmentation network can be extended from (1) as:

$$\mathcal{L}(I_s, I_t) = \sum_i \lambda_{seg}^i \mathcal{L}_{seg}^i(I_s) + \sum_i \lambda_{adv}^i \mathcal{L}_{adv}^i(I_t), \tag{5}$$

where i indicates the level used for predicting the segmentation output. We note that, the segmentation output is still predicted in each feature space, before passing through individual discriminators for adversarial learning. Hence, $\mathcal{L}_{seg}^i(I_s)$ and $\mathcal{L}_{adv}^i(I_t)$ remain in the same form as in (3) and (4), respectively. Based on (5), we optimize the following min-max criterion:

$$\max_{\mathbf{D}} \min_{\mathbf{G}} \mathcal{L}(I_s, I_t). \tag{6}$$

The ultimate goal is to minimize the segmentation loss in $\mathbf{G}$ for source images, while maximizing the probability of target predictions being considered as source predictions.

5. Network Architecture and Training

Discriminator. For the discriminator, we use an architecture similar to [31] but utilize all fully-convolutional layers to retain the spatial information. The network consists of 5 convolution layers with kernel 4×4 and stride of 2, where the channel number is {64, 128, 256, 512, 1}, respectively. Except for the last layer, each convolution layer is followed by a leaky ReLU [27] parameterized by 0.2. An up-sampling layer is added to the last convolution layer for re-scaling the output to the size of the input. We do not use any batch-normalization layers [16] as we jointly train the discriminator with the segmentation network using a small batch size.

Segmentation Network. It is essential to build upon a good baseline model to achieve high-quality segmentation results [2, 40, 42]. We adopt the DeepLab-v2 [2] framework with ResNet-101 [11] model pre-trained on ImageNet [6] as our segmentation baseline network. However, we do not use the multi-scale fusion strategy [2] due to the memory issue. Similar to the recent work on semantic segmentation [2, 40], we remove the last classification layer and modify the stride of the last two convolution layers from 2 to 1, making the resolution of the output feature maps effectively $1/8$ times the input image size. To enlarge the receptive field, we apply dilated convolution layers [40] in *conv4* and *conv5* layers with a stride of 2 and 4, respectively. After the last layer, we use the Atrous Spatial Pyramid Pooling (ASPP) [2] as the final classifier. Finally, we apply an up-sampling layer along with the softmax output to match the size of the input image. Based on this architecture, our segmentation model

Table 1. Results of adapting GTA5 to Cityscapes. We first compare our results using single-level adversarial learning in the output space with other state-of-the-art algorithms with the VGG-16 based model. Then we adopt the ResNet-101 based model and present ablation study on different components of our proposed method.

	GTA5 → Cityscapes																			
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
FCNs in the Wild [13]	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 [41]	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
CyCADA (feature) [12]	85.6	30.7	74.7	14.4	13.0	17.6	13.7	5.8	74.6	15.8	69.9	38.2	3.5	72.3	16.0	5.0	0.1	3.6	0.0	29.2
CyCADA (pixel) [12]	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
Ours (singel-level)	**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**
Baseline (ResNet)	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	**36.0**	36.6
Ours (feature)	83.7	27.6	75.5	20.3	19.9	**27.4**	28.3	**27.4**	79.0	28.4	70.1	55.1	20.2	72.9	22.5	**35.7**	**8.3**	20.6	23.0	39.3
Ours (single-level)	**86.5**	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	**75.9**	57.3	26.2	**76.3**	29.8	32.1	7.2	29.5	32.5	41.4
Ours (multi-level)	**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**

achieves 65.1% mean intersection-over-union (IoU) when trained on the Cityscapes [4] training set and tested on the Cityscapes validation set.

Multi-level Adaptation Model. We construct the above-mentioned discriminator and segmentation network as our single-level adaptation model. For the multi-level structure, we extract feature maps from the *conv4* layer and add an ASPP module as the auxiliary classifier. Similarly, a discriminator with the same architecture is added for adversarial learning. Figure 2 shows the proposed multi-level adaptation model. In this paper, we use two levels due to the balance of its efficiency and accuracy.

Network Training. To train the proposed single/multi-level adaptation model, we find that jointly training the segmentation network and discriminators in one stage is effective. In each training batch, we first forward the source image I_s to optimize the segmentation network for $\mathcal{L}_{seg}$ in (3) and generate the output P_s. For the target image I_t, we obtain the segmentation output P_t, and pass it along with P_s to the discriminator for optimizing $\mathcal{L}_d$ in (2). In addition, we compute the adversarial loss $\mathcal{L}_{adv}$ in (4) for the target prediction P_t. For the multi-level training objective in (5), we simply repeat the same procedure for each adaptation module.

We implement our network using the PyTorch toolbox on a single Titan X GPU with 12 GB memory. To train the segmentation network, we use the Stochastic Gradient Descent (SGD) optimizer 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} and is decreased using the polynomial decay with power of 0.9 as mentioned in [2]. For training the discriminator, we use the Adam optimizer [18] with the learning rate as 10^{-4} and the same polynomial decay as the segmentation network. The momentum is set as 0.9 and 0.99.

Table 2. Performance gap between the adapted model and the fully-supervised (oracle) model. We first compare results with state-of-the-art methods using the VGG based model, and then show our result using the ResNet one.

	GTA5 → Cityscapes			
method	Baseline	Adapt	Oracle	mIoU Gap
FCNs in the Wild [13]	VGG-16	27.1	64.6	-37.5
CDA [41]		28.9	60.3	-31.4
CyCADA (feature) [12]		29.2	60.3	-30.5
CyCADA (pixel) [12]		34.8	60.3	-24.9
Ours (single-level)		35.0	61.8	-25.2
Ours (multi-level)	ResNet-101	42.4	65.1	-22.7

6. Experimental Results

In this section, we present experimental results to validate the proposed domain adaptation method for semantic segmentation under different settings. First, we show evaluations of the model trained on synthetic datasets (i.e., GTA5 [32] and SYNTHIA [33]) and test the adapted model on real-world images from the Cityscapes [4] dataset. Extensive experiments including comparisons to the state-of-the-art methods and ablation study are also conducted, e.g., adaptation in the feature/output spaces and single/multi-level adversarial learning. Second, we carry out experiments on the Cross-City dataset [3], where the model is trained on one city and adapted to another city without using annotations. In all the experiments, the IoU metric is used. The code and model are available at https://github.com/wasidennis/AdaptSegNet.

6.1. GTA5

The GTA5 dataset [32] consists of 24966 images with the resolution of 1914 × 1052 synthesized from the video game based on the city of Los Angeles. The ground truth annotations are compatible with the Cityscapes dataset [4] that contains 19 categories. Following [13], we use the full set of GTA5 and adapt the model to the Cityscapes training set with 2975 images. During testing, we evaluate on the Cityscapes validation set with 500 images.

Overall Results. We present adaptation results in Table 1 with comparisons to the state-of-the-art domain adaptation methods [12, 13, 41]. For these approaches, the baseline model is trained using VGG-based architectures [24, 40]. To fairly evaluate our method, we first use the same baseline architecture (VGG-16) and train our model with the proposed single-level adaptation module. Table 1 shows that our method performs favorably against the other algorithms. While these methods all have feature adaptation modules, our results show that adapting the model in the output space achieves better performance. We note that CyCADA [12] has a pixel adaptation module by transforming source domain images to the target domain and hence obtains additional training samples. Although this strategy achieves a similar performance as ours, one can always apply pixel transformation combined with our output space adaptation to improve the results.

On the other hand, we argue that utilizing a stronger baseline model is critical for understanding the importance of different adaptation components as well as for enhancing the performance to enable real-world applications. Thus, we use the ResNet-101 based network introduced in Section 5 and train the proposed adaptation model. Table 1 shows the baseline results only trained on source images without adaptation, with comparisons to our adapted models under different settings, including feature adaptation and single/multi-level adversarial learning in the output space. Figure 3 presents some example results for adapted segmentation. We note that for small objects such as poles and traffic signs, they are harder to adapt since they easily get merged with background classes.

In addition, another factor to evaluate the adaptation performance is to measure how much gap is narrowed between the adaptation model and the fully-supervised model. Hence, we train the model using annotated ground truths in the Cityscapes dataset as the oracle results. Table 2 shows the gap under different baseline models. We observe that, although the oracle result does not differ a lot between VGG-16 and ResNet-101 based models, the gap is larger for the VGG one. It suggests us that to narrow the gap, using a deeper model with larger capacity is more practical.

Parameter Analysis. During optimizing the segmentation network **G**, it is essential to balance the weight between segmentation and adversarial losses. We first consider the single-level case in (1) and conduct experiments to observe the impact of changing λ_{adv}. Table 3 shows that a smaller λ_{adv} may not facilitate the training process significantly, while a larger λ_{adv} may propagate incorrect gradients to the network. We empirically choose λ_{adv} as 0.001 in the single-level setting.

Table 3. Sensitivity analysis of λ_{adv} for feature/output space domain adaptation in the proposed method. We show that output space adaptation can tolerate a wide range of λ_{adv}, while it is sensitive to change λ_{adv} for feature adaptation.

	GTA5 → Cityscapes			
λ_{adv}	0.0005	0.001	0.002	0.004
Feature	35.3	39.3	35.9	32.8
Output Space	40.2	41.4	40.4	40.1

Feature Level v.s. Output Space Adaptation. In the single-level setting in (1), we compare results by using feature-level or output space adaptation via adversarial learning. For feature-level adaptation, we adopt a similar strategy as used in [13, 3] and train our model accordingly. Table 1 shows that the proposed adaptation method in the output space performs better than the one in the feature level.

In addition, Table 3 shows that adaptation in the feature space is more sensitive to λ_{adv}, which causes the training process more difficult, while output space adaptation allows for a wider range of λ_{adv}. One reason is that as feature adaptation is performed in the high-dimensional space, the problem for the discriminator becomes easier. Thus, such an adapted model cannot effectively match distributions between source and target domains via adversarial learning.

Single-level v.s. Multi-level Adversarial Learning. We have shown the merits of adopting adversarial learning in the output space. In addition, we present the results of using multi-level adversarial learning in Table 1. Here, we utilize an additional adversarial module (see Figure 2) and jointly optimize (5) for two levels. To properly balance λ_{seg}^{i} and λ_{adv}^{i}, we use the same weight as in the single-level setting for the high-level output space (i.e., λ_{seg}^{1} = 1 and λ_{adv}^{1} = 0.001). Since the low-level output carries less information to predict the segmentation, we use smaller weights for both the segmentation and adversarial loss (i.e., λ_{seg}^{2} = 0.1 and λ_{adv}^{2} = 0.0002). Evaluation results show that our multi-level adversarial adaptation further improves the segmentation accuracy. More results and analysis are presented in the supplementary material.

6.2. SYNTHIA

To adapt from the SYNTHIA to Cityscapes datasets, we use the SYNTHIA-RAND-CITYSCAPES [33] set as the source domain which contains 9400 images compatible with the cityscapes annotated classes. Similar to [3], we evaluate images on the Cityscapes validation set with 13

Table 4. Results of adapting SYNTHIA to Cityscapes. We first compare our results using single-level adversarial learning in the output space with other state-of-the-art algorithms with the VGG-16 based model. Then we adopt the ResNet-101 based model and present ablation study on different components of our proposed method.

SYNTHIA → Cityscapes														
Method	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
FCNs in the Wild [13]	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 [41]	65.2	26.1	74.9	**3.7**	3.0	76.1	70.6	47.1	8.2	43.2	**20.7**	0.7	**13.1**	34.8
Cross-City [3]	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
Ours (single-level)	**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**
Baseline (ResNet)	55.6	23.8	74.6	6.1	**12.1**	74.8	79.0	**55.3**	19.1	39.6	23.3	13.7	25.0	38.6
Ours (feature)	62.4	21.9	76.3	**11.7**	11.4	75.3	80.9	53.7	18.5	59.7	13.7	20.6	24.0	40.8
Ours (single-level)	79.2	37.2	**78.8**	9.9	10.5	**78.2**	80.5	53.5	19.6	67.0	29.5	**21.6**	31.3	45.9
Ours (multi-level)	**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**

Table 5. Performance gap between the adapted model and the fully-supervised (oracle) model. We first compare results with state-of-the-art methods using the VGG based model, and then show our result using the ResNet one.

SYNTHIA → Cityscapes				
Method	Baseline	Adapt	Oracle	mIoU Gap
FCNs in the Wild [13]	VGG-16	22.9	73.8	-50.9
CDA [41]		34.8	69.6	-34.8
Cross-City [3]		35.7	73.8	-38.1
Ours (single-level)		37.6	68.4	-30.8
Ours (multi-level)	ResNet-101	46.7	71.7	-25.0

classes. For the weight in (1) and (5), we use the same ones as in the case of GTA5 dataset.

Table 4 shows evaluation results of the proposed algorithm against the state-of-the-art methods [3, 13, 41] that use feature adaptation. Similar to the experiments with the GTA5 dataset, we first utilize the same VGG-based model and train our single-level adaptation model for fair comparisons. The experimental results suggest that adapting the model in the output space performs better. Second, we compare results using different components of the proposed method with the ResNet based model. We show that the multi-level adaptation module improves the results over the baseline, feature space adaptation and single-level adaptation models. In addition, we present comparisons of mean IoU gap between adapted and oracle results in Table 5. Our method achieves the smallest gap and is the only one that can minimize the gap below 30%.

6.3. Cross-City Dataset

In addition to the synthetic-to-real adaptation for a larger domain gap, we conduct experiment on the Cross-City dataset [3] with smaller domain gaps between cities. The dataset contains four different cities: Rio, Rome, Tokyo and Taipei, in which each city has 3200 images without annotations and 100 images with pixel-level ground truths for 13 classes. Similar to [3], we use the Cityscapes training set as the source domain and adapt it to each target city using 3200 images, while 100 annotated images are used for evaluation. Since a smaller domain gap results in smaller output differences, we use smaller weights for the adversarial loss (i.e., $\lambda^i_{adv} = 0.0005$) when training our models, while the weights for segmentation remain the same as previous experiments.

We show our results in Table 6 with comparisons to [3] and our baseline models under different settings. Again, our final multi-level model achieves consistent improvement for different cities, which demonstrates the advantages of the proposed adaptation method in the output space. Note that the state-of-the-art method [3] uses a different baseline model, and we present it as a reference to analyze how much the proposed algorithm can improve.

7. Concluding Remarks

In this paper, we exploit the fact that segmentations are structured outputs and share many similarities between source and target domains. We tackle the domain adaptation problem for semantic segmentation via adversarial learning in the output space. To further enhance the adapted model, we construct a multi-level adversarial network to effectively perform output space domain adaptation at different feature levels. Experimental results show that the proposed method performs favorably against numerous baseline models and the state-of-the-art algorithms. We hope that our proposed method can be a generic adaptation model for a wide range of pixel-level prediction tasks.

Acknowledgments. W.-C. Hung is supported in part by the NSF CAREER Grant #1149783, gifts from Adobe and NVIDIA. We especially thank Yan-Ting Liu for helping ex-

Table 6. Results of adapting Cityscapes to the Cross-City dataset. We construct our baseline model using the ResNet-101 architecture, and compare results between feature adaptation and our multi-level adaptation method in the output space.

		Cityscapes → Cross-City												
City	Method	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
Rome	Cross-City [3]	79.5	29.3	84.5	0.0	22.2	80.6	82.8	29.5	13.0	71.7	37.5	25.9	1.0	42.9
	Our Baseline	**83.9**	**34.3**	87.7	13.0	**41.9**	84.6	92.5	37.7	**22.4**	80.8	38.1	39.1	5.3	50.9
	Ours (feature)	78.8	28.6	85.5	16.6	40.1	85.3	79.6	42.4	20.7	79.6	**58.8**	45.5	6.1	51.4
	Ours (output space)	**83.9**	34.2	**88.3**	**18.8**	40.2	**86.2**	**93.1**	**47.8**	21.7	**80.9**	47.8	**48.3**	**8.6**	**53.8**
Rio	Cross-City [3]	74.2	43.9	79.0	2.4	7.5	77.8	69.5	39.3	10.3	67.9	**41.2**	27.9	10.9	42.5
	Our Baseline	**76.6**	**47.3**	82.5	**12.6**	22.5	77.9	86.5	43.0	19.8	**74.5**	36.8	29.4	16.7	48.2
	Ours (feature)	73.7	44.2	83.0	6.1	18.1	79.6	86.9	51.0	22.1	73.7	31.4	**48.3**	**28.4**	49.7
	Ours (output space)	76.2	44.7	**84.6**	9.3	**25.5**	**81.8**	**87.3**	**55.3**	**32.7**	74.3	28.9	43.0	27.6	**51.6**
Tokyo	Cross-City [3]	**83.4**	**35.4**	72.8	12.3	12.7	77.4	64.3	42.7	21.5	64.1	**20.8**	8.9	40.3	42.8
	Our Baseline	82.9	31.3	**78.7**	14.2	24.5	81.6	89.2	48.6	33.3	70.5	7.7	11.5	45.9	47.7
	Ours (feature)	81.5	30.8	76.6	15.3	20.2	82.0	84.0	49.4	33.3	70.5	4.5	24.3	**51.6**	48.0
	Ours (output space)	81.5	26.0	77.8	**17.8**	**26.8**	**82.7**	**90.9**	**55.8**	**38.0**	**72.1**	4.2	**24.5**	50.8	**49.9**
Taipei	Cross-City [3]	78.6	28.6	80.0	13.1	7.6	68.2	82.1	16.8	9.4	60.4	34.0	26.5	9.9	39.6
	Our Baseline	**83.5**	**33.4**	**86.6**	12.7	**16.4**	77.0	**92.1**	17.6	**13.7**	70.7	37.7	44.4	18.5	46.5
	Ours (feature)	82.1	31.9	84.1	25.7	13.2	**77.2**	81.2	28.1	12.0	67.0	35.8	43.5	20.9	46.6
	Ours (output space)	81.7	29.5	85.2	**26.4**	15.6	76.7	91.7	**31.0**	12.5	**71.5**	**41.1**	**47.3**	**27.7**	**49.1**

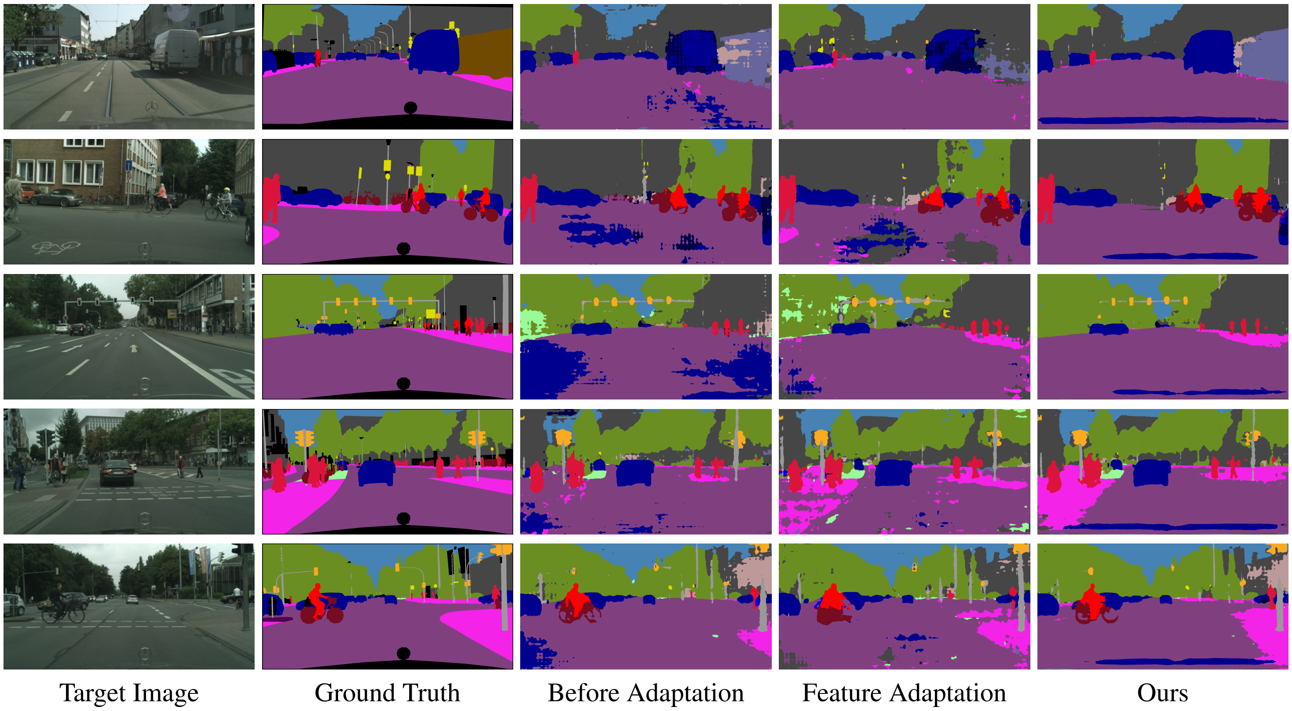

Figure 3. Example results of adapted segmentation for GTA5-to-Cityscapes. For each target image, we show results before adaptation, with feature adaptation and our adapted segmentations in the output space.

Table 7. Results of adapting GTA5 to Cityscapes.

	GTA5 → Cityscapes																		
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Vanilla-GAN	86.5	25.9	79.8	22.1	20.0	23.6	33.1	**21.8**	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	**7.2**	**29.5**	**32.5**	41.4
LS-GAN	**91.4**	**48.4**	**81.2**	**27.4**	**21.2**	**31.2**	**35.3**	16.1	**84.1**	**32.5**	**78.2**	**57.7**	**28.2**	**85.9**	**33.8**	**43.5**	0.2	23.9	16.9	**44.1**

Table 8. Results of adapting SYNTHIA to Cityscapes. mIoU and mIoU* are averaged over 16 and 13 categories, respectively.

	SYNTHIA → Cityscapes																	
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU	mIoU*
Vanilla-GAN	79.2	37.2	78.8	**10.5**	**0.3**	**25.1**	**9.9**	**10.5**	78.2	80.5	53.5	19.6	67.0	29.5	**21.6**	31.3	39.5	45.9
LS-GAN	**84.0**	**40.5**	**79.3**	10.4	0.2	22.7	6.5	8.0	**78.3**	**82.7**	**56.3**	**22.4**	**74.0**	**33.2**	18.9	**34.9**	**40.8**	**47.6**

Table 9. Results of adapting Synscapes to Cityscapes.

	Synscapes → Cityscapes																		
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Without Adaptation	81.8	40.6	76.1	23.3	16.8	36.9	36.8	40.1	83.0	34.8	84.9	59.9	37.7	78.5	20.4	20.5	7.8	27.3	52.5	45.3
Vanilla-GAN	**94.2**	**60.9**	**85.1**	29.1	25.2	38.6	**43.9**	40.8	**85.2**	29.7	88.2	**64.4**	40.6	85.8	**31.5**	43.0	28.3	**30.5**	**56.7**	52.7
LS-GAN	**94.2**	60.5	85.0	**29.2**	**25.6**	**39.8**	43.4	**43.8**	**85.2**	**35.9**	**88.3**	63.2	**41.1**	**87.2**	30.8	**44.2**	**29.8**	28.5	53.7	**53.1**

periments using LS-GAN and the Synscapes dataset, as in the appendix.

References

[1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017. 2, 3

[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. *CoRR*, abs/1606.00915, 2016. 1, 2, 4, 5

[3] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. F. Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017. 1, 2, 3, 5, 6, 7, 8

[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 *CVPR*, 2016. 2, 5, 6

[5] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[7] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 2

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. In *JMLR*, 2016. 1, 2, 4

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4

[12] 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. *CoRR*, abs/1711.03213, 2017. 3, 5, 6

[13] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 1, 3, 5, 6, 7

[14] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*, 2015. 2

[15] W.-C. Hung, Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. Scene parsing with global context embedding. In *ICCV*, 2017. 2

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[17] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 2

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[20] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015. 4

[21] G. Lin, C. Shen, A. van dan Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 1

[22] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016. 2

[23] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 1

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 6

[25] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1, 2, 4

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016. 2

[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 4

[28] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 10

[29] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 2

[30] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 2, 3

[31] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 4

[32] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2, 5

[33] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 2, 5, 6

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[35] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *ICCV*, 2017. 2

[36] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. Deep image harmonization. In *CVPR*, 2017. 1

[37] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 2

[38] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 2

[39] M. Wrenninge and J. Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *CoRR*, abs/1810.08705, 2018. 10

[40] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 1, 2, 4, 6

[41] Y. Zhang, P. David, and B. Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017. 5, 6, 7

[42] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2, 4

[43] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 1

[44] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017. 3

A. Least Squares Objective

To analyze the impact of different type of GANs in our framework, we adopt the least-squares loss function in [28] that claims to generate higher-quality results and perform more stably during GAN training. The loss for discriminator training, similar to (2), can be written as:

$$\mathcal{L}_d^{LS}(P) = \sum_{h,w} z\left(\mathbf{D}(P)^{(h,w,1)} - 1\right)^2 + (1-z)\left(\mathbf{D}(P)^{(h,w,0)}\right)^2, \tag{7}$$

where $z = 0$ if the sample is drawn from the target domain, and $z = 1$ for the sample from the source domain. Similar to (4), the adversarial loss can be written as:

$$\mathcal{L}_{adv}^{LS}(I_t) = \sum_{h,w}\left(\mathbf{D}(P_t)^{(h,w,1)} - 1\right)^2. \tag{8}$$

We use the single-level adaptation network and the ResNet-101 backbone as in the main paper, and all the other details are the same. Results on Cityscapes using GTA5 and SYNTHIA as the source domain are presented in Table 7 and Table 8, respectively. We compare the performance of the vanilla GAN (as in the main paper) and the least-squares (LS) GAN. Both tables show that using the LS-GAN objective achieves a higher mean IoU.

B. Synscapes

The Synscapes dataset [39] is a photorealistic synthetic dataset for street scene parsing. It consists of $25,000$ RGB images at 1440×720 resolution. The ground truth annotation adopts the Cityscapes convention that contains 19 categories. To adapt from Synscapes to Cityscapes, we use the entire Synsacpes dataset as the source domain. In Table 9, we show results without adaptation, with vanilla GAN, and LS-GAN, using the single-level adaptation network and the ResNet-101 backbone.

Since the domain gap between Cityscapes and Synscapes is smaller than the case using either GTA5 or SYNTHIA as the source domain, the performance without adaptation already achieves a mean IoU of 45.3%. By further using output space adaptation, the vanilla and LS GAN objectives improve the results and perform competitively.