Domain Adaptation for Structured Output via Discriminative Patch Representations

Yi-Hsuan Tsai[1] Kihyuk Sohn[1*] Samuel Schulter[1] Manmohan Chandraker[1,2]
[1]NEC Laboratories America [2]University of California, San Diego

Abstract

Predicting structured outputs such as semantic segmentation relies on expensive per-pixel annotations to learn supervised models like convolutional neural networks. However, models trained on one data domain may not generalize well to other domains without annotations for model fine-tuning. To avoid the labor-intensive process of annotation, we develop a domain adaptation method to adapt the source data to the unlabeled target domain. We propose to learn discriminative feature representations of patches in the source domain by discovering multiple modes of patch-wise output distribution through the construction of a clustered space. With such representations as guidance, we use an adversarial learning scheme to push the feature representations of target patches in the clustered space closer to the distributions of source patches. In addition, we show that our framework is complementary to existing domain adaptation techniques and achieves consistent improvements on semantic segmentation. Extensive ablations and results are demonstrated on numerous benchmark datasets with various settings, such as synthetic-to-real and cross-city scenarios.

1. Introduction

With the availability of large-scale annotated datasets [8], deep learning has made a significant impact on many computer vision tasks, such as object recognition [14, 21], detection [11], or semantic segmentation [3]. Unfortunately, learned models may not generalize when evaluated on a test domain different from the labeled training data [45]. Unsupervised domain adaptation (UDA) [10, 32] has been proposed to close the performance gap introduced by the mismatch between the source domain, where labeled data is available, and the target domain. UDA circumvents an expensive data annotation process by utilizing only unlabeled data from the target domain. Along this line, numerous UDA methods have been developed and successfully applied for classification tasks [1, 10, 23, 24, 32, 40, 41].

*Now at Google Cloud AI.

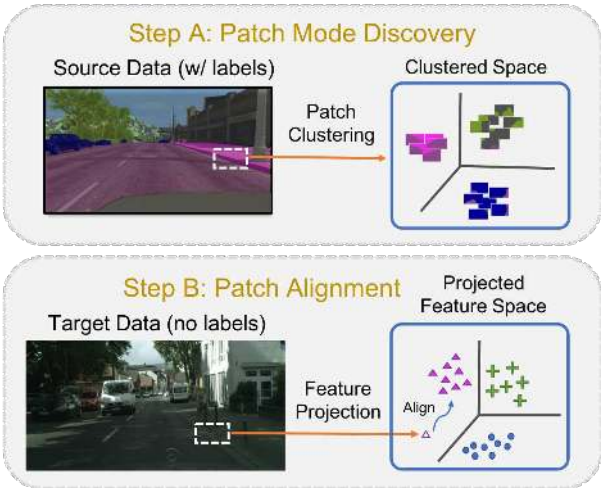


Figure 1. Our method aims at improving output distribution alignment via: 1) patch mode discovery from the source patch annotations to construct a clustered space and project to a feature space, and 2) patch alignment from the target patch representation (unfilled symbol) to the source distribution (solid symbols).

UDA is even more crucial for pixel-level prediction tasks such as semantic segmentation as annotation is prohibitively expensive. A prominent approach towards domain adaptation for semantic segmentation is distribution alignment by adversarial learning [13, 10], where the alignment may happen at different representation layers, such as pixel-level [16, 48], feature-level [16, 17] or output-level [39]. Despite these efforts, discovering all modes of the data distribution is a key challenge for domain adaptation [38], akin to difficulties also faced by generative tasks [2, 26].

A critical step during adversarial training is the use of a convolutional discriminator [19, 16, 39] that classifies patches into source or target domains. However, the discriminator is not supervised to capture several modes in the data distribution and it may end up learning only low-level differences such as tone or texture across domains. In addition, for the task of semantic segmentation, it is important to capture and adapt high-level patterns given the highly structured output space.

In this work, we propose an unsupervised domain adap-

tation method that explicitly discovers many modes in the structured output space of semantic segmentation to learn a better discriminator between the two domains, ultimately leading to a better domain alignment. We leverage the pixel-level semantic annotations available in the source domain, but instead of directly working on the output space [39], our adaptation happens in two stages. First, we extract patches from the source domain, represent them using their annotation maps and discover major modes by applying K-means clustering, which groups patches into K clusters (Step A in Figure 1). Each patch in the source domain can now be assigned to a ground truth cluster or mode index. We then introduce a K-way classifier that predicts the cluster or mode index of each patch, which can be supervised in the source domain but not in the target domain.

Second, different from the output space alignment [39], our method, referred as patch-level alignment (Step B in Figure 1) operates on the K-dimensional probability vector space after projecting to the clustered space that already discovers various patch modes. This is in contrast to prior art that operates on either pixel- [48], feature- [16] or output-level [39]. The learned discriminator on the clustered space can back-propagate the gradient through the cluster or mode index classifier to the semantic segmentation network.

In experiments, we follow the setting of [16] and perform pixel-level road-scene semantic segmentation. We experiment under various settings, including synthetic-to-real (GTA5 [30], SYNTHIA [31] to Cityscapes [7]) and cross-city (Cityscapes to Oxford RobotCar [25]) adaptation. We provide an extensive ablation study to validate each component in the proposed framework. Our approach is also complementary to existing domain adaptation techniques, which we demonstrate by combining with output space adaptation [39], pixel-level adaptation [15] and pseudo label re-training [50]. Our results show that the learned representations improve segmentation results consistently and achieve state-of-the-art performance.

Our contributions are summarized as follows. First, we propose an adversarial adaptation framework for structured prediction that explicitly tries to discover and predict modes of the output patches. Second, we demonstrate the complementary nature of our approach by integration into three existing domain adaptation methods, which can all benefit from it. Third, we extensively analyze our approach and show state-of-the-art results on various domain adaptation benchmarks for semantic segmentation.[1]

2. Related Work

We discuss unsupervised domain adaptation methods for image classification and pixel-level structured prediction tasks, and works on learning disentangled representations.

[1]The project page is at www.nec-labs.com/~mas/adapt-seg.

UDA for Image Classification. UDA methods have been developed for classification by aligning the feature distributions between the source and the target domains. Conventional methods use hand-crafted features [9, 12] to minimize the discrepancy across domains, while recent algorithms utilize deep architectures [10, 40] to learn domain-invariant features. One common practice is to adopt adversarial learning [10] or to minimize the Maximum Mean Discrepancy [23]. Several variants have been developed by designing different classifiers [24] and loss functions [40, 41], and for distance metric learning [36, 37]. In addition, other recent work aims to enhance feature representations by pixel-level transfer [1] and maximum classifier discrepancy [33].

UDA for Semantic Segmentation. Following the practice in image classification, domain adaptation for pixel-level predictions has been studied. [16] introduces to tackle the semantic segmentation problem for road-scene images by adapting from synthetic images via aligning global feature representations. In addition, a category-specific prior, e.g., object size and class distribution is extracted from the source domain and is transferred to the target distribution as a constraint. Instead of designing such constraints, [46] applies the SVM classifier to capture label distributions on superpixels as the property to train the adapted model. Similarly, [6] proposes a class-wise domain adversarial alignment by assigning pseudo labels to the target data.

More recently, numerous approaches are proposed to improve the adapted segmentation and can be categorized as follows: 1) output space [39] and spatial-aware [5] adaptations aim to align the global structure (e.g., scene layout) across domains; 2) pixel-level adaptation synthesizes target samples [15, 27, 43, 47] to reduce the domain gap during training the segmentation model; 3) pseudo-label re-training [34, 50] generates pseudo ground truth of target images to finetune the model trained on the source domain. While the most relevant approaches to ours are from the first category, they do not handle intrinsic domain gaps such as camera poses. In contrast, the proposed patch-level alignment is able to match patches at various image locations across domains. We also note that, the other two categories or other techniques such as robust loss function design [49] are orthogonal to the contribution of this work. In Section 4.3, we show that our patch-level representations can be integrated with other domain adaptation methods to further enhance performance.

Learning Disentangled Representations. Learning a latent disentangled space has led to a better understanding for numerous tasks such as facial recognition [29], image generation [4, 28], and view synthesis [22, 44]. These approaches use predefined factors to learn interpretable representations of the image. [22] propose to learn graphic codes that are disentangled with respect to various image transformations, e.g.,

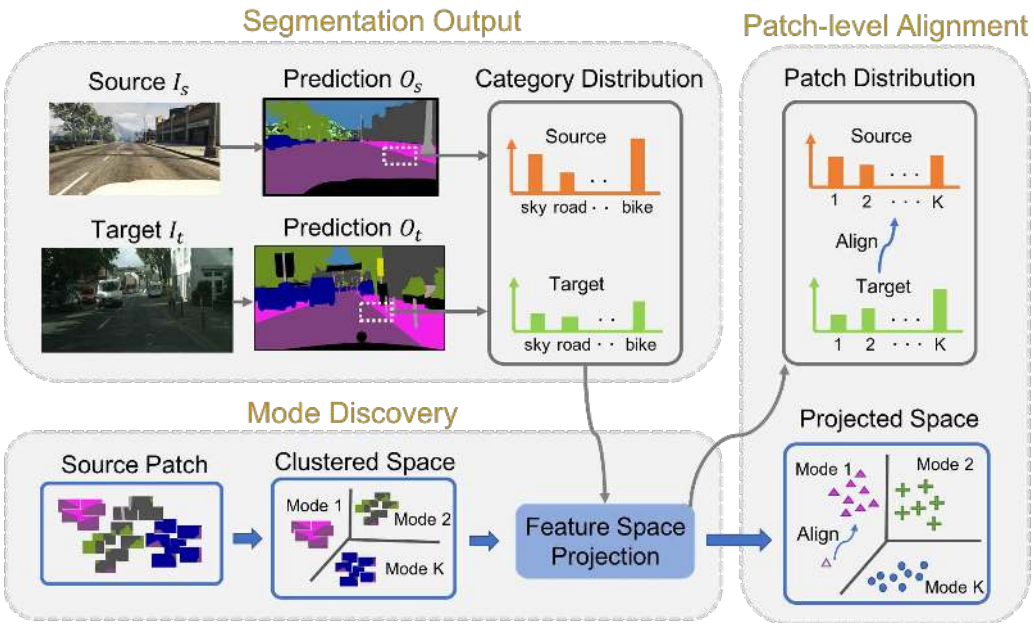


Figure 2. An overview of our patch-level alignment. For our method, the category distribution is projected to the patch distribution through a clustered space that is constructed by discovering K patch modes in the source domain. For the target data, we then align patch distributions across domains using adversarial learning in this K-dimensional space. In comparison, note that output space adaptation methods only have a step that directly aligns category distributions without considering multiple modes in the source data.

pose and lighting, for rendering 3D images. Similarly, [44] synthesize 3D objects from a single image via an encoder-decoder architecture that learns latent representations based on the rotation factor. Recently, AC-GAN [28] develops a generative adversarial network (GAN) with an auxiliary classifier conditioned on the given factors such as image labels and attributes.

Although these methods present promising results on using the specified factors and learning a disentangled space to help the target task, they focus on handling data in a single domain. Motivated by this line of research, we propose to learn discriminative representations for patches to help the domain adaptation task. To this end, we take advantage of the available label distributions and naturally utilize them as a disentangled factor, in which our framework does not need to predefine any factors like conventional methods.

3. Domain Adaptation for Structured Output

In this section, we describe our framework for predicting structured outputs: an adversarial learning scheme to align distributions across domains by using discriminative output representations of patches.

3.1. Algorithm Overview

Given the source and target images $I_s, I_t \in \mathbb{R}^{H\times W\times 3}$, where only the source data is annotated with per-pixel semantic categories Y_s, we seek to learn a semantic segmentation model $\mathbf{G}$ that works on both domains. Since the target domain is unlabeled, our goal is to align the predicted output distribution O_t of the target data with the source distribution O_s, which is similar to [39]. However, such distribution is not aware of the local difference in patches and thus is not able to discover a diverse set of modes during adversarial learning. To tackle this issue, and in contrast to [39], we project the category distribution of patches to the clustered space that already discovers various patch modes (*i.e.*, K clusters) based on the annotations in the source domain. For target data, we then employ adversarial learning to align the patch-level distributions across domains in the K-dimensional space.

3.2. Patch-level Alignment

As in Figure 2, we seek for ways to align patches in a clustered space that provides a diverse set of patch modes. One can also treat this procedure as learning prototypical output representations of patches by clustering ground truth segmentation annotations from the source domain. In what follows, we introduce how we construct the clustered space and learn discriminative patch representations. Then we describe adversarial alignment using the learned patch representation. The detailed architecture is shown in Figure 3.

Patch Mode Discovery. To discover modes and learn a discriminative feature space, class labels [35] or predefined factors [28] are usually provided as supervisory signals. However, it is non-trivial to assign a class membership to

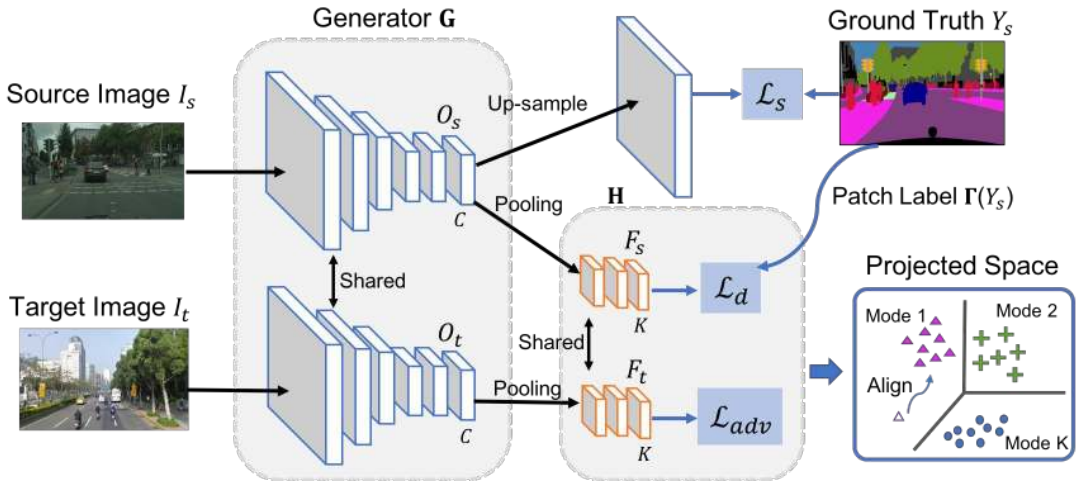


Figure 3. The proposed network architecture that consists of a generator $\mathbf{G}$ and a categorization module $\mathbf{H}$ for learning discriminative patch representations through 1) patch mode discovery supervised by the patch classification loss $\mathcal{L}_d$, and 2) patch-level alignment using the adversarial loss $\mathcal{L}_{adv}$. In the projected space, solid symbols denote source representations and unfilled ones are target representations pulled to the source distribution.

individual patches of an image. One may apply unsupervised clustering of image patches, but it is unclear whether the constructed clustering would separate patches in a semantically meaningful way. In this work, we make use of per-pixel annotations available in the source domain to construct a space of semantic patch representation. To achieve this, we use label histograms for patches. We first randomly sample patches from source images, use a 2-by-2 grid on patches to extract spatial label histograms, and concatenate them to obtain a $2 \times 2 \times C$ dimensional vector. Second, we apply K-means clustering on these histograms, thereby assigning each ground truth label patch a unique cluster index. We define the process of finding the cluster membership for each patch in a ground truth label map Y_s as $\Gamma(Y_s)$.

To incorporate this clustered space for training the segmentation network $\mathbf{G}$ on source data, we add a classification module $\mathbf{H}$ on top of the predicted output O_s, which tries to predict the cluster membership $\Gamma(Y_s)$ for all locations. We denote the learned representation as $F_s = \mathbf{H}(\mathbf{G}(I_s)) \in (0,1)^{U \times V \times K}$ through the softmax function, where K is the number of clusters. Here, each data point on the spatial map F_s corresponds to a patch of the input image, and we obtain the group label for each patch via $\Gamma(Y_s)$. Then the learning process to construct the clustered space can be formulated as a cross-entropy loss:

$$\mathcal{L}_d(F_s, \Gamma(Y_s); \mathbf{G}, \mathbf{H}) = -\sum_{u,v} \sum_{k \in K} CE^{(u,v,k)}, \quad (1)$$

where $CE^{(u,v,k)} = \Gamma(Y_s)^{(u,v,k)} \log(F_s^{(u,v,k)})$.

Adversarial Alignment. The ensuing task is to align the representations of target patches to the clustered space constructed in the source domain, ideally aligned to one of the K modes. To this end, we utilize an adversarial loss between F_s and F_t, where F_t is generated in the same way as described above. Note that, the patch-level feature F is now transformed from the category distribution O to the clustered space defined by K-dimensional vectors. We then formulate the patch distribution alignment in an adversarial objective:

$$\mathcal{L}_{adv}(F_s, F_t; \mathbf{G}, \mathbf{H}, \mathbf{D}) = \sum_{u,v} \mathbb{E}[\log \mathbf{D}(F_s)^{(u,v,1)}] + \mathbb{E}[\log(1 - \mathbf{D}(F_t)^{(u,v,1)})], \quad (2)$$

where $\mathbf{D}$ is the discriminator to classify whether the feature representation F is from the source or the target domain.

Learning Objective. We integrate (1) and (2) into the min-max problem (for clarity, we drop all arguments to losses except the optimization variables):

$$\min_{\mathbf{G},\mathbf{H}} \max_{\mathbf{D}} \mathcal{L}_s(\mathbf{G}) + \lambda_d \mathcal{L}_d(\mathbf{G}, \mathbf{H}) + \lambda_{adv} \mathcal{L}_{adv}(\mathbf{G}, \mathbf{H}, \mathbf{D}), \quad (3)$$

where $\mathcal{L}_s$ is the supervised cross-entropy loss for learning the structured prediction (e.g., semantic segmentation) on source data, and λ's are the weights for different losses.

3.3. Network Optimization

To solve the optimization problem in Eq. (3), we follow the procedure of training GANs [13] and alternate two steps: 1) update the discriminator $\mathbf{D}$, and 2) update the networks $\mathbf{G}$ and $\mathbf{H}$ while fixing the discriminator.

Update the Discriminator D. We train the discriminator $\mathbf{D}$ to classify whether the feature representation F is from the source (labeled as 1) or the target domain (labeled as

0). The maximization problem with respect to $\mathbf{D}$ in (3) is equivalent to minimizing the binary cross-entropy loss:

$$\mathcal{L}_{\mathbf{D}}(F_s, F_t; \mathbf{D}) = -\sum_{u,v} \log(\mathbf{D}(F_s)^{(u,v,1)}) + \log(1 - \mathbf{D}(F_t)^{(u,v,1)}). \quad (4)$$

Update the Networks G and H. The goal of this step is to push the target distribution closer to the source distribution using the optimized $\mathbf{D}$, while maintaining good performance on the main tasks using $\mathbf{G}$ and $\mathbf{H}$. As a result, the minimization problem in (3) is the combination of two supervised loss functions with the adversarial loss, which can be expressed as a binary cross-entropy function that assigns the source label to the target distribution:

$$\mathcal{L}_{\mathbf{G},\mathbf{H}} = \mathcal{L}_s + \lambda_d \mathcal{L}_d - \lambda_{adv} \sum_{u,v} \log(\mathbf{D}(F_t)^{(u,v,1)}). \quad (5)$$

We note that updating $\mathbf{H}$ also influences $\mathbf{G}$ through back-propagation, and thus the feature representations are enhanced in $\mathbf{G}$. In addition, we only require $\mathbf{H}$ during the training phase, so that runtime for inference is unaffected compared to the output space adaptation approach [39].

3.4. Implementation Details

Network Architectures. The generator consists of the network $\mathbf{G}$ with a categorization module $\mathbf{H}$. For a fair comparison, we follow the framework used in [39] that adopts DeepLab-v2 [3] with the ResNet-101 architecture [14] as our baseline network $\mathbf{G}$. To add the module $\mathbf{H}$ on the output prediction O, we first use an adaptive average pooling layer to generate a spatial map, where each data point on the map has a desired receptive field corresponding to the size of extracted patches. Then this pooled map is fed into two convolution layers and a feature map F is produced with the channel number K. Figure 3 illustrates the main components of the proposed architecture. For the discriminator $\mathbf{D}$, input data is a K-dimensional vector and we utilize 3 fully-connected layers similar to [41], with leaky ReLU activation and channel numbers $\{256, 512, 1\}$.

Implementation Details. We implement the proposed framework using the PyTorch toolbox on a single Titan X GPU with 12 GB memory. To train the discriminators, we use the Adam optimizer [20] with initial learning rate of 10^{-4} and momentums set as 0.9 and 0.99. For learning the generator, we use the Stochastic Gradient Descent (SGD) solver where the momentum is 0.9, the weight decay is 5×10^{-4} and the initial learning rate is 2.5×10^{-4}. For all the networks, we decrease the learning rates using the polynomial decay with a power of 0.9, as described in [3]. During training, we select $\lambda_d = 0.01$, $\lambda_{adv} = 0.0005$ and $K = 50$ fixed for all the experiments. Note that we first train the model only using the loss $\mathcal{L}_s$ for 10K iterations to avoid initially noisy predictions and then train the network using all the loss functions. More details of the hyper-parameters such as image and patch sizes are provided in the supplementary material.

4. Experimental Results

We evaluate the proposed framework for domain adaptation on semantic segmentation. We first conduct an extensive ablation study to validate key components of our algorithm. Second, we show that the proposed method can be integrated with various domain adaptation techniques, including output space adaptation [39], pixel-level adaptation [15], and pseudo label re-training [50]. This demonstrates that our learned patch-level representations are complementary to a wide range of domain adaptation strategies and provide additional benefits. Finally, we present a hybrid model that performs favorably against state-of-the-art approaches on numerous benchmark datasets and settings.

4.1. Evaluated Datasets and Metric

We evaluate our domain adaptation method on semantic segmentation under various settings, including synthetic-to-real and cross-city. First, we adapt the synthetic GTA5 [30] dataset to the Cityscapes [7] dataset that contains real road-scene images. Similarly, we use the SYNTHIA [31] dataset, which has a larger domain gap to Cityscapes images. For these experiments, we follow [16] to split data into training and test sets. As another example with high practical impact, we apply our method on data captured in different cities and weather conditions by adapting Cityscapes with sunny images to the Oxford RobotCar [25] dataset containing rainy scenes. We manually select 10 sequences in the Oxford RobotCar dataset tagged as "rainy" and randomly split them into 7 sequences for training and 3 for testing. We sequentially sample 895 images for training and annotate 271 images with per-pixel semantic segmentation ground truth as the test set for evaluation. The annotated ground truths are made publicly available at the project page. For all experiments, Intersection-over-Union (IoU) ratio is used as the evaluation metric.

4.2. Ablation Study and Analysis

In Table 1, we conduct the ablation study and analysis of the proposed patch-level alignment on the GTA5-to-Cityscapes scenario to understand the impact of different loss functions and design choices in our framework.

Loss Functions. In Table 1, we show different steps of the proposed method, including the model without adaptation, using discriminative patch features and the final patch-level alignment. Interestingly, we find that adding discriminative

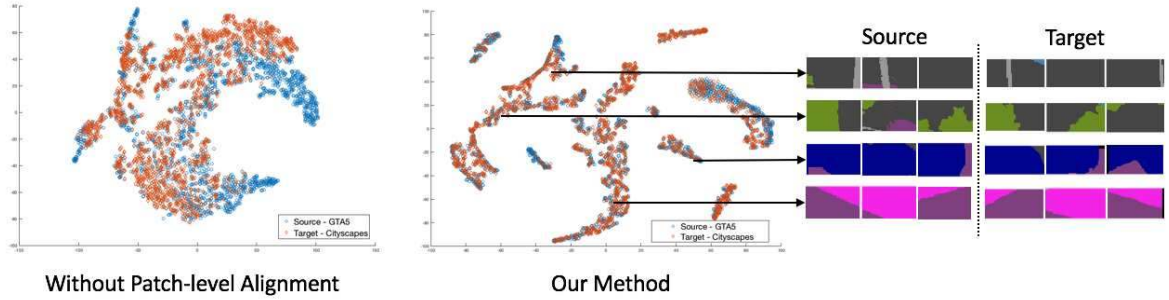


Figure 4. Visualization of patch-level representations. We first show feature representations for our method using t-SNE and compare to the baseline without the proposed patch-level alignment. In addition, we show patch examples in the clustered space. In each group, patches are similar in appearance (each color represents a semantic label) between the source and target domains.

Table 1. Ablation study of the proposed loss functions on GTA5-to-Cityscapes using the ResNet-101 network.

GTA5 → Cityscapes		
Method	Loss Func.	mIoU
Without Adaptation	$\mathcal{L}_s$	36.6
Discriminative Feature	$\mathcal{L}_s + \mathcal{L}_d$	38.8
Patch-level Alignment	$\mathcal{L}_s + \mathcal{L}_d + \mathcal{L}_{adv}$	41.3

patch representations without any alignments ($\mathcal{L}_s + \mathcal{L}_d$) already improves the performance (from 36.6% to 38.8%), which demonstrates that the learned feature representation enhances the discrimination and generalization ability. Finally, the proposed patch-level adversarial alignment improves the mIoU by 4.7%.

Impact of Learning Clustered Space. K-means provides an additional signal to separate different patch patterns, while performing alignment in this clustered space. Without the clustered loss L_d, it would be difficult to align patch modes across two domains. To validate it, we run an experiment by only using L_s and L_{adv} but removing L_d, and the performance is reduced by 1.9% compared to our method (41.3%). This shows the importance of learning the clustered space supervised by the K-means process.

Impact of Cluster Number K. In Figure 5, we study the impact of the cluster number K used to construct the patch representation, showing that the performance is robust to K. However, when K is too large, e.g., larger than 300, it would cause confusion between patch modes and increases the training difficulty. To keep both efficiency and accuracy, we use $K = 50$ throughout the experiments.

Visualization of Feature Representations. In Figure 4, we show the t-SNE visualization [42] of the patch-level features in the clustered space of our method and compare with the one without patch-level adaptation. The result shows that with adaptation in the clustered space, the features are

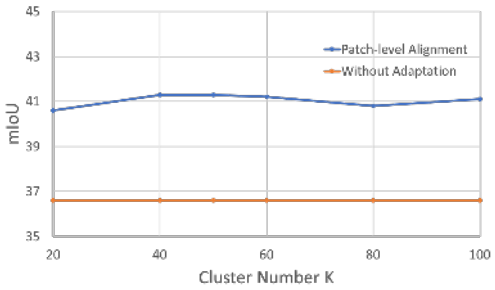


Figure 5. The performance of our method with respect to different numbers of clusters K on GTA5-to-Cityscapes.

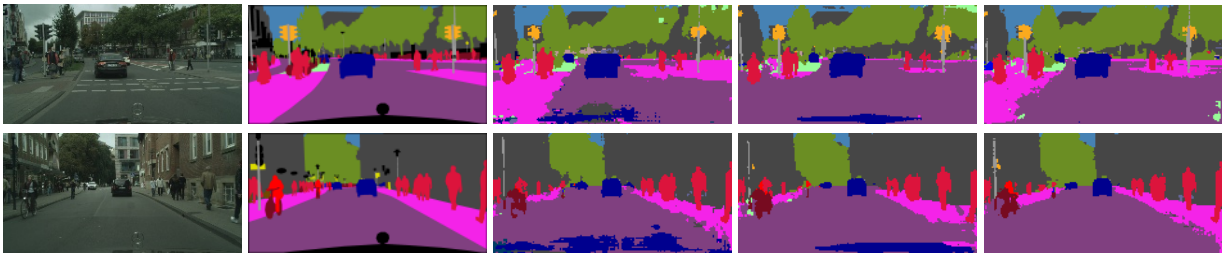
embedded into groups and the source/target representations overlap well. In addition, we present example source/target patches with high similarity.

4.3. Improvement on Domain Adaptation Methods

The learned patch representation via the proposed patch alignment enhances feature representations and is complementary to various DA methods, which we demonstrate by combining with output-space adaptation (Ou), pixel-level adaptation (Pi) and pseudo label re-training (Ps). Our results show consistent improvement in all cases, *e.g.*, 1.8% to 2.7% on GTA5-to-Cityscapes, as shown in Table 2.

Output Space Adaptation. We first consider methods that align the *global* layout across domains as in [5, 39]. Our proposed cluster prediction network $\mathbf{H}$ and the corresponding loss $\mathcal{L}_{adv}$ can be simply added into [39]. Since these methods only align the global structure, adding our method helps figuring out local details better and improves the segmentation quality.

Pixel-level Adaptation. We utilize CyCADA [15] as the pixel-level adaptation algorithm and produce synthesized images in the target domain from source images. To train our model, we add synthesized samples into the labeled training set with the proposed patch-level alignment. Note that, since synthesized samples share the same pixel-level



Target Image | Ground Truth | Before Adaptation | Output Alignment | Patch Alignment

Figure 6. Example results for GTA5-to-Cityscapes. Our method often generates the segmentation with more details (e.g., sidewalk and pole) while producing less noisy regions compared to the output space adaptation approach [39].

Table 2. Performance improvements in mIoU of integrating our patch-level alignment with existing domain adaptation approaches on GTA5-to-Cityscapes using the ResNet-101 network.

GTA5 → Cityscapes (19 Categories)			
Methods	Base	+ Patch-Alignment	Δ
Without Adaptation	36.6	41.3	+4.7
(Ou)tput Space Ada.	41.4	43.2	+1.8
(Pi)xel-level Ada.	42.2	44.9	+2.7
(Ps)eudo-GT	41.8	44.2	+2.4
(Fu)sion	44.5	46.5	+2.0

Table 3. Performance improvements in mIoU of integrating our patch-level alignment with existing domain adaptation approaches on SYNTHIA-to-Cityscapes using the ResNet-101 network.

SYNTHIA → Cityscapes (16 Categories)			
Methods	Base	+ Patch-Alignment	Δ
Without Adaptation	33.5	37.0	+3.5
(Ou)tput Space Ada.	39.5	39.9	+0.4
(Pi)xel-level Ada.	35.8	37.0	+1.2
(Ps)eudo-GT	37.4	38.9	+1.5
(Fu)sion	37.9	40.0	+2.1

annotations as the source data, they can be also considered in our clustering process and the optimization in (3).

Pseudo-label Re-training. Pseudo-label re-training is a natural way to improve the segmentation quality in domain adaptation [50] or semi-supervised learning [18]. The end-to-end trainable framework [18] uses an adversarial scheme to identify self-learnable regions, which makes it an ideal candidate to integrate our patch-level adversarial loss.

Results and Discussions. The results for combining the proposed patch-level alignment with the three above mentioned DA methods are shown in Tables 2 and 3 for GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes, respectively. We can observe that adding patch-level alignment improves in all cases. For reference, we also show the gain from adding patch-level alignment to the plain segmentation network (without adaptation). Even when combining all three DA methods, i.e., Fusion (Fu), the proposed patch-alignment further improves the results significantly ($\geq 2.0\%$). Note that, the combination of all DA methods including patch alignment, *i.e.*, Fu + Patch-Alignment, achieves the best performance in both cases.

As a comparison point, we also try to combine pixel-level adaptation with output space alignment (Pi + Ou), but the performance is 0.7% worse than ours, *i.e.*, Pi + Patch-Alignment, showing the advantages of adopting patch-level alignment. On SYNTHIA-to-Cityscapes in Table 3, we find that Pi and Ps are less effective than Ou, likely due to the poor quality of the input data in the source domain, which also explains the lower performance of the combined model (Fu). This also indicates that directly combining different DA methods may not improve the performance incrementally. However, adding the proposed patch-alignment improves the results consistently in all settings.

4.4. Comparisons with State-of-the-art Methods

We have validated that the proposed patch-level alignment is complementary to existing domain adaptation methods on semantic segmentation. In the following, we compare our final model (Fu + Patch-Alignment) with state-of-the-art algorithms under various scenarios, including synthetic-to-real and cross-city cases.

Synthetic-to-real Case. We first present experimental results for adapting GTA5 to Cityscapes in Table 4. We utilize two different architectures, i.e., VGG-16 and ResNet-101, and compare with state-of-the-art approaches via feature adaptation [16, 46], pixel-level adaptation [15], pseudo label re-training [50] and output space alignment [5, 39]. We show that the proposed framework improves over existing methods by 2.5% and 5.1% in mean IoU for two architectures, respectively. In Table 5, we present results for adapting SYNTHIA to Cityscapes and similar improvements are observed compared to state-of-the-arts. In addition, we shows visual comparisons in Figure 6 and more results are presented in the supplementary material.

Cross-city Case. Adapting between real images across different cities and conditions is an important scenario for

Table 4. Results of adapting GTA5 to Cityscapes. The first and second groups adopt VGG-16 and ResNet-101 networks, respectively.

	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 [16]	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 [46]	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
ST [50]	83.8	17.4	72.1	14.3	2.9	16.5	16.0	6.8	**81.4**	24.2	47.2	40.7	7.6	71.7	10.2	7.6	0.5	11.1	0.9	28.1
CBST [50]	66.7	26.8	73.7	14.8	9.5	**28.3**	25.9	10.1	75.5	15.7	51.6	47.2	6.2	71.9	3.7	2.2	**5.4**	**18.9**	**32.4**	30.9
CyCADA [15]	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
Output Space [39]	**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
Ours (VGG-16)	**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**
Without Adaptation	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
Feature Space [39]	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
Road [5]	76.3	36.1	69.6	28.6	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
Output Space [39]	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 (ResNet-101)	**92.3**	**51.9**	**82.1**	**29.2**	**25.1**	24.5	**33.8**	**33.0**	**82.4**	32.8	**82.2**	**58.6**	**27.2**	**84.3**	**33.4**	**46.3**	2.2	**29.5**	32.3	**46.5**

Table 5. Results of adapting SYNTHIA to Cityscapes. The first and second groups adopt VGG-16 and ResNet-101 networks, respectively. 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*
FCNs in the Wild [16]	11.5	19.6	30.8	**4.4**	0.0	20.3	0.1	**11.7**	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	20.2	22.1
CDA [46]	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	29.0	34.8
Cross-City [6]	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
ST [50]	0.2	14.5	53.8	1.6	0.0	18.9	0.9	7.8	72.2	80.3	**48.1**	6.3	67.7	4.7	0.2	4.5	23.9	27.8
Output Space [39]	**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
Ours (VGG-16)	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**
Without Adaptation	55.6	23.8	74.6	9.2	0.2	24.4	6.1	**12.1**	74.8	79.0	**55.3**	19.1	39.6	23.3	13.7	25.0	33.5	38.6
Feature Space [39]	62.4	21.9	76.3	**11.5**	0.1	24.9	**11.7**	11.4	75.3	80.9	53.7	18.5	59.7	13.7	20.6	24.0	35.4	40.8
Output Space [39]	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
Ours (ResNet-101)	**82.4**	**38.0**	78.6	8.7	**0.6**	**26.0**	3.9	11.1	75.5	**84.6**	53.5	**21.6**	**71.4**	**32.6**	19.3	**31.7**	**40.0**	**46.5**

practical applications. We choose a challenging case where the weather condition is different (i.e., sunny v.s. rainy) in two cities by adapting Cityscapes to Oxford RobotCar. The proposed framework achieves a mean IoU of 72.0% averaged on 9 categories, significantly improving the model without adaptation by 10.1%. To compare with the output space adaptation method [39], we run the code released by the authors and obtain a mean IoU of 69.5%, which is 2.5% lower than the proposed method. Further results and comparisons are provided in the supplementary material.

5. Conclusions

In this paper, we present a domain adaptation method for structured output via patch-level alignment. We propose to learn discriminative representations of patches by constructing a clustered space of the source patches and adopt an adversarial learning scheme to push the target patch distributions closer to the source ones. With patch-level alignment, our method is complementary to various domain adaptation approaches and provides additional improvement. We conduct extensive ablation studies and experiments to validate the effectiveness of the proposed method under numerous challenges on semantic segmentation, including synthetic-to-real and cross-city scenarios, and show that our approach performs favorably against previous methods.

References

[1] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In

CVPR, 2017. 1, 2

[2] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *ICLR*, 2017. 1

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. 1, 5

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016. 2

[5] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, 2018. 2, 6, 7, 8

[6] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017. 2, 8

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 5

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[9] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013. 2

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *JMLR*, 2016. 1, 2

[11] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 1

[12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 2

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[15] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 2, 5, 6, 7, 8

[16] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 1, 2, 5, 7, 8

[17] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *ECCV*, 2018. 1

[18] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018. 7

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[22] Tejas D. Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015. 2

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1, 2

[24] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016. 1, 2

[25] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000km: The oxford robotcar dataset. *The International Journal of Robotics Research (IJRR)*, 36(1), 2017. 2, 5

[26] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017. 1

[27] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018. 2

[28] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 2, 3

[29] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML*, 2014. 2

[30] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2, 5

[31] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 2, 5

[32] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 1

[33] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 2

[34] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M. Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *ECCV*, 2018. 2

[35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 3

[36] Kihyuk Sohn, Sifei Liu, Guangyu Zhong, Xiang Yu, Ming-Hsuan Yang, and Manmohan Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *ICCV*, 2017. 2

[37] Kihyuk Sohn, Wenling Shang, Xiang Yu, and Manmohan Chandraker. Unsupervised domain adaptation for distance metric learning. In *ICLR*, 2019. 2

[38] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019. 1

[39] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 2, 3, 5, 6, 7, 8

[40] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 1, 2

[41] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1, 2, 5

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 6

[43] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *ECCV*, 2018. 2

[44] Jimei Yang, Scott Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NIPS*, 2015. 2, 3

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 1

[46] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017. 2, 7, 8

[47] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, 2018. 2

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017. 1, 2

[49] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *ECCV*, 2018. 2

[50] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 2, 5, 7, 8