Pixel-level Intra-domain Adaptation for Semantic Segmentation

Zizheng Yan[13], Xianggang Yu[13], Yipeng Qin[4], Yushuang Wu[13], Xiaoguang Han[13*], Shuguang Cui[23]
[1]SSE, CUHK-Shenzhen [2] FNii, CUHK-Shenzhen [3] Shenzhen Research Institute of Big Data
[4]Cardiff University
{zizhengyan,xianggangyu,yushuangwu}@link.cuhk.edu.cn
{hanxiaoguang,shuguangcui}@cuhk.edu.cn,qiny16@cardiff.ac.uk

ABSTRACT

Recent advances in unsupervised domain adaptation have achieved remarkable performance on semantic segmentation tasks. Despite such progress, existing works mainly focus on bridging the inter-domain gaps between the source and target domain, while only few of them noticed the intra-domain gaps within the target data. In this work, we propose a pixel-level intra-domain adaptation approach to reduce the intra-domain gaps within the target data. Compared with image-level methods, ours treats each pixel as an instance, which adapts the segmentation model at a more fine-grained level. Specifically, we first conduct the inter-domain adaptation between the source and target domain; Then, we separate the pixels in target images into the easy and hard subdomains; Finally, we propose a pixel-level adversarial training strategy to adapt a segmentation network from the easy to the hard subdomain. Moreover, we show that the segmentation accuracy can be further improved by incorporating a continuous indexing technique in the adversarial training. Experimental results show the effectiveness of our method against existing state-of-the-art approaches.

CCS CONCEPTS

• **Computing methodologies** → **Scene understanding**; **Image segmentation**.

KEYWORDS

Semantic segmentation; Deep unsupervised domain adaptation; Intra-domain adaptation

ACM Reference Format:
Zizheng Yan, Xianggang Yu, Yipeng Qin, Yushuang Wu, Xiaoguang Han, and Shuguang Cui. 2021. Pixel-level Intra-domain Adaptation for Semantic Segmentation. In *Proceedings of the 29th ACM Int'l Conference on Multimedia (MM '21), Oct. 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475174

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '21, October 20–24, 2021, Virtual Event, China.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475174

1 INTRODUCTION

Semantic segmentation is born to be a pixel-level task that aims to assign a class label to each pixel in an image. Compared to other related tasks like image classification and object detection, semantic segmentation is more fine-grained and thus has gained popularity in various applications, *e.g.* autonomous driving [9, 21], image synthesis and manipulation [27, 51], and medical imaging [29]. Despite their superiority, modern semantic segmentation models build on the data-hungry deep learning techniques and thus suffer from the costly pixel-wise annotation of training data. For cost reduction's sake, both researchers and practitioners turned to the synthetic data that are automatically annotated upon generation [28, 30]. However, segmentation models trained on synthetic data usually generalize poorly to real-world data due to the synthetic-to-real domain gaps. To bridge such domain gaps, unsupervised domain adaptation (UDA) was introduced to minimize the distributional discrepancy between synthetic and real-world data [4, 6, 19, 32, 36, 39, 52].

Recently, Pan *et al.* [26] observed that the domain gaps do not only exist between the synthetic and real-world data (*a.k.a.* inter-domain gap) but also exist between different pieces of the real-world data (*a.k.a.* intra-domain gap). Based on this observation, they proposed a two-step UDA method for semantic segmentation. Similar to other methods, they first train an inter-domain model by adapting a segmentation network from the synthetic to real-world data. While unlike other methods, they further separate the real-world data into the "easy" and "hard" subdomains according to how easily an image can be segmented. Then, an *intra-domain* model is trained by adapting another segmentation network from the "easy" to "hard" subdomain.

Although being state-of-the-art, IntraDA [26] has a key shortcoming that we would like to improve upon: the separation of the easy and hard subdomains at **image-level** is not sufficient for the intra-domain adaptation because class labels are assigned at **pixel-level**. Specifically, IntraDA [26] determines the easiness of segmentation by averaging the entropy of class probabilities across all pixels in the segmentation map predicted by the inter-domain adapted segmentation network. As a result, "hard" pixels in an easy image and "easy" pixels in a hard image are both ignored, *e.g.* , in Cityscapes [8] training set, when $\lambda = 0.67$[1], approximately 16% of the pixels in easy subdomain images have entropy values larger than the threshold, which means that these 16% pixels are hard but misclassified into the easy subdomain. In this work, we

[1]Pan *et al.* [26] first rank the target images based on the mean values of their corresponding entropy maps. Then, they set the 100λth percentile mean entropy value as the threshold to separate easy and hard images, where $\lambda \in [0, 1]$ is a hyperparameter used to control the proportion of easy images. Target images with mean entropy values smaller than the threshold are classified as easy images and vice versa.

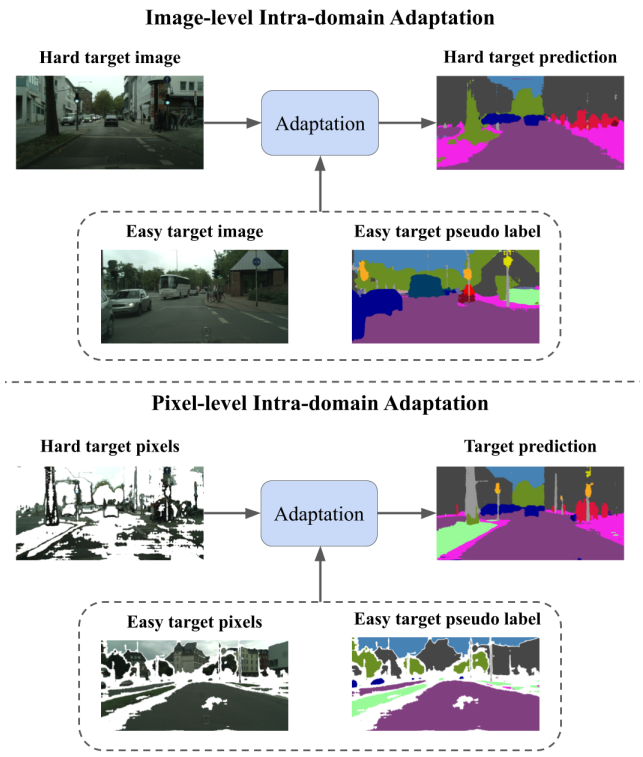


Figure 1: Image-level intra-domain adaptation [26] splits target domain images into easy and hard subdomains and performs adaptation in-between. However, our pixel-level intra-domain adaptation splits pixels of target domain images into easy and hard subdomains and thus achieves better segmentation accuracy by performing the adaptation at a more fine-grained level.

address the aforementioned problem by performing intra-domain adaptation at **pixel-level** (see Figure 1). Specifically, our pixel-level intra-domain adaptation method consists of two steps: 1) Separating the easy and hard subdomains at pixel-level according to the predictions of a pre-trained inter-domain adapted segmentation network; 2) Adapting another segmentation network from the easy to hard subdomain at pixel-level. Note that we also modify the adversarial training in step 2 with a continuous indexing technique to further improve the segmentation accuracy. Experimental results show that our method achieves higher segmentation accuracy than state-of-the-art intra-adaptation methods on benchmark datasets.

Our Contributions. First, we identify a key shortcoming of the image-level intra-domain adaptation method: "hard" pixels in an easy image and "easy" pixels in a hard image are both ignored. Second, we propose a pixel-level intra-domain adaptation method to address the aforementioned shortcoming. Third, we show that the segmentation accuracy can be further improved by incorporating a continuously indexed adversarial training method.

2 RELATED WORK

Unsupervised Domain Adaptive Segmentation. The goal of unsupervised domain adaptive segmentation is to train a segmentation network that achieves good performance in an unlabeled target domain when only the source domain data are annotated. Methodology-wise, existing methods build on three techniques: 1) adversarial learning [7, 22, 32, 33, 36, 42, 44, 46], 2) image-to-image translation [2, 10, 12, 14, 17, 25, 31, 40] and 3) self-training [1, 15, 18, 19, 24, 36, 43, 47, 49, 53].

Adversarial learning employs a discriminator to mitigate the distributional discrepancy between the source and target domain in the feature and/or output space of the segmentation network. For example, [46] encouraged the segmentation network to learn domain-invariant features by incorporating a discriminator to distinguish between the feature maps of the source and target data; [32] used a discriminator to align the output distributions of the source and target data, *i.e.*the softmax probability maps generated by the segmentation network. Improving upon [32], Vu *et al.* [36] borrowed ideas from self-training and showed that aligning the distributions in a transformed output space (*i.e.*the entropy space) is more advantageous. Observing that the gaps of foreground and background classes are of different sizes between the source and target domain, [39] refined the feature alignment process by treating the foreground and background classes differently.

Image-to-image translation based methods assume that the domain gaps are mostly in the low-level features (*e.g.* textures) rather than the semantic structures of an image and proposed to close them by translating images between the source and target domain. For example, CyCADA [12] translates images from the source to target domain by enforcing the cycle-consistency [50] that preserves semantic labels. In addition to cycle-consistency, [6] augmented the regularization of the network with more consistency constraints, *e.g.* cross domain consistency. Addressing the imbalanced data sizes between the source and target domain, [41] proposed to invert the translation direction to target-to-source and enforce the semantic consistency by incorporating a cycle-reconstruction loss.

Self-training trains a segmentation network by exploiting the pseudo labels of target domain images and their associated confidence values. For example, [53] formulates self-training as a joint learning process of both the model and the pseudo labels in an "easy-to-hard" way. [36] penalizes low-confidence predictions by minimizing an entropy loss over target data. Observing that the gradient magnitudes are imbalanced during entropy minimization [36], Chen *et al.*[4] employed a maximum square loss to alleviate the problem. [19, 39] set thresholds on confidence values and only use high-confidence pseudo labels in self-training. Similarly, [49] uses an uncertainty estimation module to rectify the noisy pseudo labels. As a result, only low-uncertainty (*i.e.*high-confidence) pseudo labels are used in the training.

Domain Separation. Most existing methods take the separation of source and target domains for granted: according to the data collection process, it is straightforward to separate the source and target domains by the scenarios where the data are collected. [37] pointed out that the above separation strategy is not always optimal. For instance, in medical applications, one needs to adapt

disease diagnosis models across patients of different ages, blood pressure levels, activity levels, etc. Thus, it is more appropriate to divide both the source and target domains into subdomains according to these continuous variables. Apart from that, a recent work IntraDA [26] shows that utilizing the predictions of a pre-trained segmentation network, traditionally-defined target domains can be further separated into "easy" and "hard" subdomains by ranking how "easy" an image can be segmented. After the separation, *intra-domain adaption* is performed to improve the segmentation accuracy by directly applying existing domain adaption techniques (*e.g.* self-training) to the two subdomains to close their domain gaps. However, by analyzing the results of IntraDA, we observed a key shortcoming: ranking the easiness of segmentation at **image-level** is not sufficient for intra-domain adaptation because pseudo labels are predicted at **pixel-level**.

In this work, we address the aforementioned inconsistency by ranking the easiness of segmentation at pixel-level and separate the subdomains accordingly. Since existing domain adaption techniques are all at image-level and cannot be used without modification, we also developed new methods to adapt the segmentation network from the "easy" to "hard" subdomain at pixel-level.

3 SYSTEM PIPELINE

In this section, we revisit the general pipeline of the two-stage domain adaptation method [26], which consists of i) the inter-domain adaptive segmentation [32, 36] and ii) the intra-domain adaptive segmentation.

3.1 Inter-Domain Adaptation

Let $\mathcal{D}_s = (x_j^s, y_j^s)_{j=1}^{N_s}$ be the source domain, $\mathcal{D}_t = (x_i)_{i=1}^{N}$ be the unlabelled target domain, where $x_j^s, x_i \in \mathbb{R}^{H\times W\times 3}$ are input images, $y_j^s \in \{0,1\}^{H\times W\times|C|}$ is the pixel-level semantic annotations of x_j^s and C is the set of class index. The goal of inter-domain adaptation is to train a segmentation network G_{inter} that achieves good performance on $\mathcal{D}_t$ using the labeled data in $\mathcal{D}_s$ and the unlabelled data in $\mathcal{D}_t$. Following the adversarial learning approach [32], this is implemented by i) training a discriminator D_{inter} to classify whether $G_{inter}(x_i)$ is from the source or target domain; ii) training the segmentation network G_{inter} to fool D_{inter}. Accordingly, the loss functions can be formally written as:

$$\begin{aligned}\mathcal{L}_{G_{inter}}(x_j^s, y_j, x_i) = &-\sum_{h,w}\sum_{c}(y_j^s)^{(h,w,c)}\log G_{inter}(x_j^s)^{(h,w,c)}\\ &-\log\left[D_{inter}(G_{inter}(x_i))\right],\\ \mathcal{L}_{D_{inter}}(x_j^s, x_i) = &-\log\left[1-D_{inter}(G_{inter}(x_i)\right]\\ &-\log\left[D_{inter}(G_{inter}(x_j^s))\right].\end{aligned} \tag{1}$$

3.2 Intra-Domain Adaptation

Observing that there are still domain gaps within the target domain $\mathcal{D}_t$, Pan *et al.* [26] proposed to reduce them by separating $\mathcal{D}_t$ into two subdomains, namely the easy and hard subdomains, and adapting another segmentation network G from the easy to hard subdomain. Specifically, they first train an inter-domain adaptation model by the method in [36], and then rank each image $x_i \in \mathcal{D}_t$ by the mean value of its predicted entropy map $I_i \in \mathbb{R}^{H\times W}$. By setting thresholds on the ranking, x_i is classified into the easy subdomain if its predicted entropy is low and vice versa. Finally, they adapt G from the easy to hard subdomain using only the predicted labels of easy subdomain images (*i.e.*, pseudo labels).

As aforementioned, the image-level mean value of I_i is not sufficient for capturing the pixel-level domain gaps among predicted labels. To this end, we propose a method to separate the subdomains at pixel-level and also a pixel-level adversarial training strategy to adapt G from the easy to hard subdomain.

4 PIXEL-LEVEL INTRA-DOMAIN ADAPTATION

Following the two-stage pipeline described in section 3, we first train an inter-domain adapted segmentation network G_{inter} and then construct a set $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$ with N samples, where $x_i \in \mathbb{R}^{H\times W\times 3}$ is an input image from target domain and $y_i = G_{inter}(x_i) \in [0,1]^{H\times W\times|C|}$ is its corresponding "soft segmentation map" of $|C|$ classes. Given $\mathcal{D}$, this section shows: i) how to separate the *pixels* of x_i into an easy and hard split according to y_i; ii) how to adapt the training of an intra-domain segmentation network G from the easy to the hard subdomain at *pixel-level*. An overview of our method is shown in Figure 2.

4.1 Subdomain Separation at Pixel-Level

Let $x_i^{(h,w)}$ be the pixel at position (h,w) of x_i, we sort it into the easy and the hard subdomains by thresholding its confidence values $y_i^{(h,w,k)}$: the easier a pixel can be classified into a category c, the higher its confidence value is. Addressing the challenge of imbalanced classes, we refine the proposed thresholding method to be class-wise and represent the results with a binary mask $M_i \in \{0,1\}^{H\times W}$ that:

$$M_i^{(h,w)} = \begin{cases} 1, & \forall k \in C, y_i^{(h,w,k)} \le \alpha\cdot t_k \\ 0, & \exists k \in C, y_i^{(h,w,k)} > \alpha\cdot t_k \end{cases}, \tag{2}$$

where t_k is the threshold of class k, $\alpha \in [0,1]$ is a scaling parameter to control the separation, $M_i^{(h,w)} = 1$ means that the pixel belongs to the hard subdomain and vice versa. Specifically, t_k is determined by the median of all confidence values of class k across the dataset [19, 39]: t_k is set to 0.9 when the median value is larger than 0.9; Otherwise, it is set to the median value. According to the separation matrix M_i, we assign pseudo labels $\hat{y}_i^{(h,w)}$ to corresponding pixels $x_i^{(h,w)}$ as:

$$\hat{y}_i^{(h,w)} = \begin{cases} \underset{k\in C}{\text{argmax}}\ \mathbb{1}_{[y_i^{(h,w,k)} > \alpha\cdot t_k]}(y_i^{(h,w,k)}), & M_i^{(h,w)} = 0 \\ none, & M_i^{(h,w)} = 1 \end{cases}, \tag{3}$$

where $\mathbb{1}(\cdot)$ is a function that returns the input if the condition is satisfied; Otherwise, a "none" value is returned and thus excluded from the domain of the argmax function. In addition, we calculate the proportion of easy pixels as:

$$\text{Prop.} = \frac{1}{HWN}\sum_{i}^{N}\mathbf{1}_H^\top(J - M_i)\mathbf{1}_W, \tag{4}$$

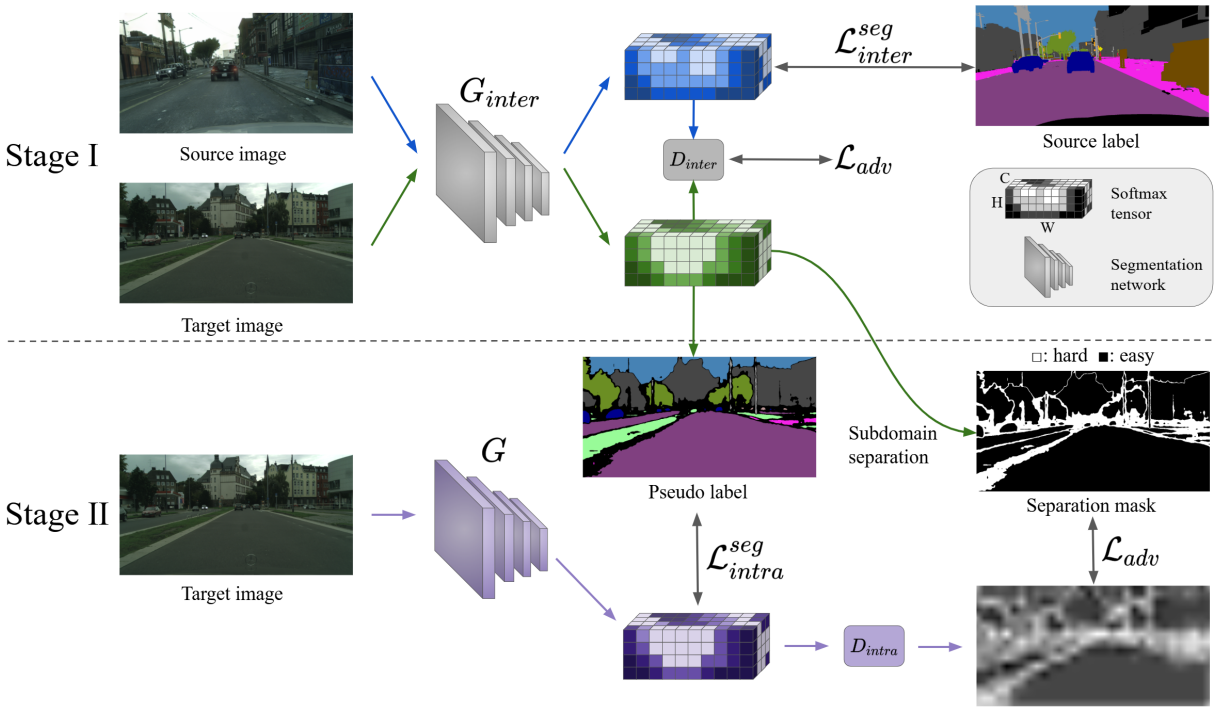


Figure 2: Pipeline of the proposed domain adaptation method. Stage I: inter-domain adaptation. In this stage, we train an inter-domain adaptive segmentation model G_{inter} using an arbitrarily-selected existing method, *e.g.* [19, 32]. Stage II: pixel-level intra-domain adaptation. First, we separate the pixels in the target image into the easy and hard subdomains according to the predictions of G_{inter}. These two subdomains are represented by a binary mask M (Eq. 2), where the black and white pixels belong to the easy and hard subdomains respectively. Then, we train an intra-domain adaptive segmentation model G by i) matching easy subdomain pixels with the corresponding pseudo labels (Eq. 3) predicted by G_{inter} (Eq. 5); ii) performing pixel-level adversarial learning on hard subdomain pixels to make them indistinguishable from the easy subdomain ones (Eq. 6, 8).

where N is number of samples, $\mathbf{1}_H$ and $\mathbf{1}_W$ are vectors of ones with size $H \times 1$ and $W \times 1$ respectively, J is a matrix of ones with size $H \times W$. Note that Prop. is indirectly controlled by α via M_i.

4.2 Subdomain Adaptation at Pixel-Level

With the aid of the separation mask M_i and the pseudo label tensor $\hat{y}_i$, we propose a new domain adaption framework to enable pixel-level adaption between subdomains. Given a segmentation network G with input x_i, we adapt G from the easy to the hard subdomain by i) matching the pseudo labels in $\hat{y}_i$ that belong to the easy subdomain and ii) fooling the discriminator D which distinguishes between easy and hard *pixels* through adversarial learning.

Self-training with Partial Pseudo Labels. To avoid the noise introduced by low-confidence pixels, we only use the pseudo labels whose corresponding pixels belong to the easy subdomain, *i.e.*, $\hat{y}_i^{(h,w)} \neq none$ (Eq. 3). For simplicity, we convert $\hat{y}_i^{(h,w)}$ to its one-hot representation $\hat{y}_i^{(h,w,c)}$ and formulate the partial segmentation loss as:

$$\mathcal{L}_G^{seg}(x_i, \hat{y}_i) = -\sum_{(h,w)\in E_i} \sum_c \hat{y}_i^{(h,w,c)} \log G(x_i)^{(h,w,c)}, \quad (5)$$

where E_i is the set consisting of all (h, w) that $x_i^{(h,w)}$ belongs to the easy subdomain (*i.e.*, $M_i^{(h,w)} = 0$).

Pixel-Level Adversarial Learning. Contrasting previous adversarial learning methods [26] that discriminate between the binary (*e.g.* True and False) labels assigned to *images*, we formulate pixel-level adversarial learning by discriminating the binary labels assigned to *pixels*. To this end, we represent the pixel-wise labels of an input image x_i with M_i (Eq. 2) and employ a special discriminator architecture that outputs a matrix of the same size of M_i, *i.e.*, $H \times W$. Accordingly, our pixel-level adversarial loss functions are:

$$\begin{aligned}\mathcal{L}_D^{adv}(x_i, M_i) = &-\log\left[\mathbf{1}_H^\top (D(G(x_i)) \circ M_i)\mathbf{1}_W\right] \\ &-\log\left[\mathbf{1}_H^\top ((J - D(G(x_i))) \circ (J - M_i))\mathbf{1}_W\right], \quad (6)\\ \mathcal{L}_G^{adv}(x_i, M_i) = &-\log\left[\mathbf{1}_H^\top ((J - D(G(x_i))) \circ M_i)\mathbf{1}_W\right],\end{aligned}$$

where $\circ$ denotes the hadamard product operator.

Continuously Indexed Adversarial Learning. Although most existing methods use categorical domain labels as in Eq. 6, the predicted labels of G are essentially continuous as softmax probabilities. This implies that useful information might be lost during

the categorization of continuous labels, especially for those in the hard subdomain. Thus, inspired by [37], we directly use the predicted probabilities as continuous domain indexes for the adversarial learning. Specifically, we modify the binary label mask M_i to its continuous version Z_i using the predicted probabilities $y_i^{(h,w,k)}$ as:

$$Z_i^{(h,w)} = \begin{cases} 0, & M_i^{(h,w)} = 0 \\ 1 - \max_{k \in C}(y_i^{(h,w,k)}), & M_i^{(h,w)} = 1 \end{cases}. \tag{7}$$

Hence, the objective of the discriminator is to regress towards Z_i. However, unlike [37], we treat the pixels in the easy and hard subdomains separately when training the segmentation network G. Specifically, we apply the partial segmentation loss to pixels in the easy subdomain (Eq. 5) and the adversarial loss to pixels in the hard subdomain. Thus, our continuous adversarial loss functions are:

$$\begin{aligned} \hat{\mathcal{L}}_D^{adv}(x_i, Z_i) &= \|D(G(x_i)) - Z_i\|_F^2, \\ \hat{\mathcal{L}}_G^{adv}(x_i, M_i, Z_i) &= \|D(G(x_i)) \circ M_i\|_F^2, \end{aligned} \tag{8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Incorporating the partial segmentation loss (Eq. 5), the overall loss functions for the generator G and the discriminator D are:

$$\begin{aligned} \mathcal{L}_D &= \hat{\mathcal{L}}_D^{adv}, \\ \mathcal{L}_G &= \mathcal{L}_G^{seg} + \beta \hat{\mathcal{L}}_G^{adv}, \end{aligned} \tag{9}$$

where β is a weighting parameter balancing $\mathcal{L}_G^{seg}$ and $\hat{\mathcal{L}}_G^{adv}$.

5 IMPLEMENTATION

5.1 Network Architecture

Segmentation Network. Similar to [32], we adopt the DeepLab-v2 [3] framework with a ResNet-101 [11] model pre-trained on ImageNet as the backbone. Hence, the output size is 1/8 times the input image size. Since our pixel-level training (Eq. 5, 6) requires the output size to match the input size, we bilinearly upsample the segmentation output to be of the same spatial size as the input image. Finally, we attach a softmax layer to the rear of the segmentation network to generate class-wise probabilities for each pixel.

Discriminator Network. Following [26, 32], we use two discriminators to conduct multi-level adversarial learning. One discriminator takes the final segmentation probability map as input and another one takes the probability map of the auxiliary classifier as input. The auxiliary classifier is the same as the one used in [26, 32]. To conduct pixel-level adversarial learning, similar to the segmentation network, we bilinearly upsample the outputs of both discriminators to be of the same spatial size of the input image.

5.2 Datasets

We evaluate our method on two popular synthetic-to-real adaptation scenarios: i) GTA5 [28] to Cityscapes [8] and ii) SYNTHIA [30] to Cityscapes [8].

GTA5. GTA5 [28] is a synthetic dataset which contains 24,966 images with a resolution of 1,914 × 1,052. The synthetic images are collected from a video game based on the city of Los Angeles. There are 33 categories in the ground truth annotations. Similar to [26, 32, 36], we only use the 19 categories that are compatible with the Cityscapes [8] annotations.

SYNTHIA. The SYNTHIA [30] dataset contains 9,400 synthetic urban scene images of resolution 1,280 × 760. Similar to [26, 32, 36], we only use the 13 categories that are compatible with the Cityscapes [8] annotations.

Cityscapes. Cityscapes [8] is a real world urban scene dataset that contains 3,975 images collected from different cities. Similar to [26, 32, 36], 2,975 images are selected from the training set of Cityscapes [8] for training, 500 images are selected from the evaluation set of Cityscapes [8] for evaluation.

5.3 Training Details

Inter-domain Adaptation. For GTA5→Cityscapes, we collect the source domain images by transferring the GTA5 images to the style of the Cityscapes images [19] and generate the pseudo labels by training the inter-domain adaptation model of [32]. For SYNTHIA→Cityscapes, we generate the pseudo labels using the released pre-trained model of [19].

Intra-domain Adaptation. We use the Cityscapes images and the generated pseudo labels to train our intra-domain adaptation model. The hyperparameters of the two discriminators and the auxiliary classifier are the same as the ones used in [32]. We use Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay parameter of 5×10^{-4} to optimize the network. The initial learning rate is set to be 2.4×10^{-4} and polynomially [3] decays with a power of 0.9. For the discriminator networks, Adam [16] optimizers with momentum 0.9 and 0.99 are used. The initial learning rate is 10^{-4} and decays using the same strategy as SGD. Similar to multi-round training in [19, 39, 52], we train our network for 2 rounds. For the second round training (denoted as T=1), we utilize KLD regularization [52] to prevent overfitting to pseudo labels. The batch size is 4 and multi-scale training and testing is utilized. Our training is carried out on an NVIDIA Tesla V100 GPU.

6 EXPERIMENTAL RESULTS

6.1 GTA5 to Cityscapes.

Quantitative results. In Table 1, we quantitatively compare the IoU and mean IoU (mIoU) of our method (PixIntraDA) against those of the state-of-the-art methods. INTER represents the inter-domain adaptation model used in our method. It can be observed that our method outperforms all other models with a promising mIoU of 54.2%. Moreover, the proposed method also shows superior performance in terms of the per class IoU score, especially in the minor categories (*e.g.*, "motor", "bike"), which implies that proposed method can extract more fine-grained features.

Ablation studies. Our PixIntraDA consists of four parts, pixel-level adversarial learning (PLA), continuously indexed adversarial learning (CTS), one more round training (T=1) and KLD regularization [52], among which PLA and CTS are our main contributions. We validate the effectiveness of each part, as shown in Table 3. It can be observed that our PLA method outperforms the inter-domain adaptation model (INTER) by 4.2% in mIoU. Moreover, the proposed CTS can further improve the performance.

Table 1: Semantic segmentation results of GTA5→Cityscapes. The best result in each column is highlighted in bold fonts. INTER: the inter-domain adaptation model used in our method.

GTA5 ⟶ Cityscapes																				
Method	road	sidewalk	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motor.	bike	mIoU
DCGAN [40]	85.0	30.8	81.3	25.8	21.2	22.2	25.4	26.6	83.4	36.7	76.2	58.9	24.9	80.7	29.5	42.9	2.5	26.9	11.6	41.7
AdapSegNet [32]	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
CLAN [23]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
ADVENT [36]	87.6	21.4	82.0	34.8	26.2	28.5	35.6	23.0	84.5	35.1	76.2	58.6	30.7	84.8	34.2	43.4	0.4	28.4	35.2	44.8
BDL [19]	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
AdaPatch [34]	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
MaxSquare [5]	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	**85.3**	39.4	78.2	63.0	22.9	84.6	36.4	43.0	5.5	34.7	33.5	46.4
PyCDA [20]	90.5	36.3	84.4	32.4	28.7	34.6	36.4	31.5	86.8	37.9	78.5	62.3	21.5	85.6	27.9	34.8	18.0	22.9	49.3	47.4
Diff [39]	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	**43.3**	85.3	57.0	31.5	83.8	**42.6**	48.5	1.9	30.4	39.0	49.2
CrCDA [13]	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6
FADA [38]	92.5	47.5	85.1	37.6	32.8	33.4	33.8	18.4	85.3	37.7	83.5	63.2	**39.7**	87.5	32.9	47.8	1.6	34.9	39.5	49.2
MRNet [48]	90.5	35.0	84.6	34.3	24.0	36.8	44.1	42.7	84.5	33.6	82.5	63.1	34.4	85.8	32.9	38.2	2.0	27.1	41.8	48.3
MRKLD [52]	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1
LRTIR [14]	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2
FDA [43]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
IAST [24]	94.1	**58.8**	85.4	**39.7**	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	24.7	35.2	40.2	52.2
CAG [45]	90.4	51.6	83.8	34.2	27.8	38.4	25.3	**48.4**	85.4	38.2	78.1	58.6	34.6	84.7	21.9	42.7	**41.1**	29.3	37.2	50.2
INTER (ours)	91.2	47.6	81.5	28.5	17.6	29.3	28.3	24.2	82.7	36.3	80.2	56.4	27.9	82.9	33.9	42.0	0.6	29.3	26.4	44.6
PixIntraDA (ours)	**93.4**	56.1	**85.9**	29.7	**34.3**	**39.1**	**47.8**	43.8	**86.2**	37.6	**89.2**	**68.2**	38.8	**87.8**	39.6	**57.4**	0.1	**46.4**	**49.5**	**54.2**

Table 2: Semantic segmentation results of SYNTHIA→Cityscapes. The best result in each column is highlighted in bold fonts. Note that in this experiment, BDL [19] is used as the inter-domain adaptation model in our method, *i.e.*, PixIntraDA.

SYNTHIA ⟶ Cityscapes														
Method	road	sidewalk	building	light	sign	vegetation	sky	person	rider	car	bus	motorbike	bike	mIoU
AdaptSegNet [32]	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
CLAN [23]	81.3	37.0	80.1	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	47.8
ADVENT [36]	85.6	42.2	79.7	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	48.0
IntraDA [26]	84.3	37.7	79.5	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	48.9
FADA [38]	84.5	40.1	**83.1**	4.8	27.2	**84.8**	84.0	53.5	22.6	85.4	43.7	26.8	27.8	52.5
Diff [39]	83.0	44.0	80.3	17.1	15.8	80.5	81.8	59.9	33.1	70.2	37.3	28.5	45.8	52.1
LRTIR [14]	92.6	53.2	79.2	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	49.3
FDA [43]	79.3	35.0	73.2	19.9	24.0	61.7	82.6	61.4	31.1	83.9	40.8	38.4	51.1	52.5
MRNet [48]	82.0	36.5	80.4	18.0	13.4	81.1	80.8	61.3	21.7	84.4	32.4	14.8	45.7	50.2
PyCDA [20]	75.5	30.9	83.3	27.3	33.5	84.7	85.0	64.1	25.4	85.0	45.2	21.2	32.0	53.3
MRKLD [52]	67.7	32.2	73.9	22.2	**31.2**	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	50.1
IAST [24]	81.9	41.5	83.3	30.9	28.8	83.4	85.0	65.5	30.8	**86.5**	38.2	33.1	52.7	57.0
CAG [45]	84.7	40.8	81.7	13.3	22.7	84.5	77.6	64.2	27.8	80.9	19.7	22.7	48.3	51.5
BDL [19]	**86.0**	**46.7**	80.3	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	51.4
PixIntraDA (ours)	81.5	45.8	77.8	**32.5**	23.9	83.2	**87.6**	**68.9**	**36.8**	79.6	**47.3**	**44.0**	**54.7**	**58.7**

Comparison with IntraDA [26]. As Table 4 shows, we also compare our method with IntraDA [26] under the same experimental setting. To make a fair comparison, we i) bilinearly upsample the discriminator outputs of IntraDA [26] as in our method; ii) train IntraDA [26] using the pseudo labels generated by our inter-domain adaptation model (INTER). It can be observed that our method outperforms the inter-domain adaptation model (*i.e.*, INTER) by

Table 3: Ablation study on GTA5→Cityscapes. INTER: our inter-domain adaptation model. PLA: pixel-level adversarial learning (Eq. 6). CTS: continuously indexed adversarial learning (Eq. 8). T=1: the model is trained for another round. KLD: KLD regularization [52]

Model	PLA	CTS	T=1	KLD	mIoU
INTER					45.2
+PLA	✓				49.4
+CTS	✓	✓			50.3
+T=1	✓	✓	✓		53.3
+KLD	✓	✓	✓	✓	**54.2**

Table 4: Comparison with IntraDA [26] on GTA→Cityscapes. For a fair comparison, the IntraDA [26] model is retrained using the pseudo labels generated by INTER.

Model	mIoU	mIoU gain
INTER	45.2	-
IntraDA [26]	47.4	2.2
PLA	49.4	4.2
PLA + CTS	**50.3**	**5.1**

Table 5: Analysis of the proportion of easy pixels on GTA5→Cityscapes, where λ^1 = Prop. (Eq. 4) for IntraDA [26]. Note that *Prop. = 0* means intra-domain adaptation is not applied and *Prop. = 1* means that all the pseudo labels are used in self-training (Eq. 5) and none of them are used in adversarial learning (Eq. 6 and 8).

Proportion of Easy Pixels *vs* mIoU					
Model	Prop. = 0	0.57	0.67	0.79	1
	α = -	0.7	0.85	1.00	-
IntraDA [26]	45.2	47.1	47.4	46.9	46.0
PLA	45.2	49.0	49.4	49.1	46.0
PLA + CTS	45.2	**49.3**	**50.3**	**49.9**	46.0

4.2%-5.1% in mIoU, while IntraDA [26] only outperforms INTER by 2.2%.

Qualitative results. To get an intuitive understanding, we visualize the segmentation results of both IntraDA [26] and our method in Figure 3. Facilitating the comparison, we highlight the superiority of our method with colored boxes.

Analysis of the proportion of easy pixels. Similar to [26], we conduct an experiment on the proportion of easy pixels, as shown in Table 5. It can be observed that our method outperforms IntraDA [26] on all the three non-degenerate cases, *i.e.*, Prop. $\in (0, 1)$. Similar to IntraDA [26], our method achieves its best performance when Prop. = 0.67.

Versatility. As Table 6 shows, we justify the versatility of our method by testing its performance with a different pseudo label thresholding scheme: instance adaptive selection (IAS) [24]. Instead of determining a global set of per-class thresholds for all samples in the dataset (section 4.1), IAS [24] applies a unique set of per-class thresholds to each sample, which is determined by a convex combination of the current sample thresholds and a historical one during pseudo label generation. It can be observed that our method can also achieve a significant performance gain when IAS is used.

Table 6: Justification of the versatility of our method on GTA→Cityscapes. For PLA + IAS and PLA + CTS + IAS, we assign pseudo labels using instance adaptive selector (IAS) [24] rather than our pseudo label thresholding scheme (Eq. 3).

Model	mIoU	mIoU gain
INTER	45.2	-
PLA + IAS	49.0	3.8
PLA + CTS + IAS	**49.5**	**4.3**

Table 7: Ablation study on SYNTHIA→Cityscapes. We use the pretrained model provided by BDL [19] as the inter-domain adaptation model.

Model	PLA	CTS	T=1	KLD	mIoU
BDL [19]					51.4
+ PLA	✓				54.5
+ CTS	✓	✓			55.1
+ T=1	✓	✓	✓		57.5
+ KLD	✓	✓	✓	✓	**58.7**

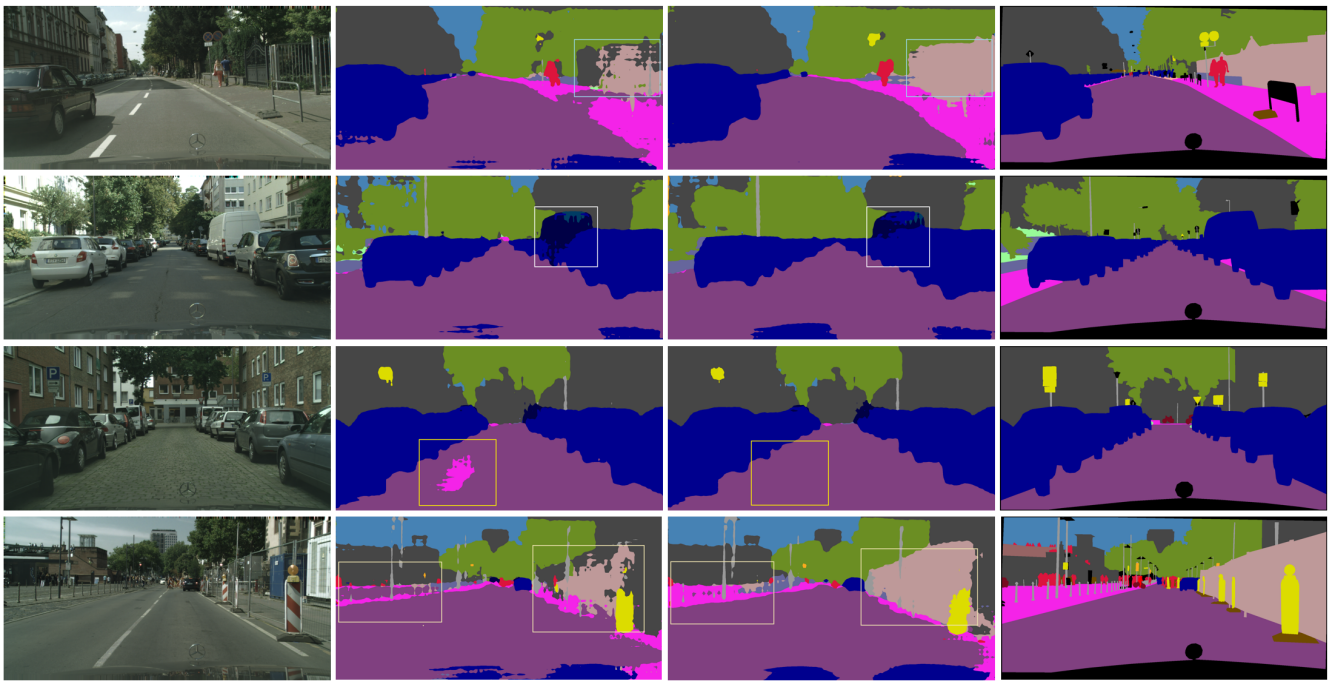
Table 8: Analysis of the proportion of easy pixels on SYNTHIA→Cityscapes. Note that *Prop. = 0* means intra-domain adaptation is not applied and *Prop. = 1* means that all the pseudo labels are used in self-training (Eq. 5) and none of them are used in adversarial learning (Eq. 6 and 8).

Proportions of Easy Pixels *vs* mIoU				
Model	Prop. = 0	0.66	0.87	1
	α = -	0.79	1.00	-
PLA	51.4	54.5	54.2	52.7
PLA + CTS	51.4	**55.1**	**54.7**	52.7

Feature visualization. We use t-SNE [35] to visualize the final features generated by the backbone network of IntraDA [26] and our method (PixIntraDA). As Figure 4 shows, IntraDA [26] can produce separated features, yet it is still hard for linear classification. In comparison, our method produces i) more dispersed feature clusters and ii) more compact intra-class features, which is more amenable to classification.

6.2 SYNTHIA to Cityscapes

Quantitative Results. Following [19, 23, 39], we evaluate the IoU and mIoU of all the 13 classes shared between SYNTHIA and Cityscapes. As Table 2 shows, our method outperforms the baseline (*i.e.*, BDL [19]) by 7.3% in mIoU, which implies that our model achieves the new state-of-the-art performance of SYNTHIA→Citys-



(a) Input (b) IntraDA (c) PixIntraDA (ours) (d) Ground Truth

Figure 3: Qualitative results on GTA5→Cityscapes. (a) and (d) are the images and their corresponding ground truth annotations from the Cityscapes validation set. (b) and (c) are the segmentation results of IntraDA [26] and our method respectively. The superiority of our method is highlighted with colored boxes.

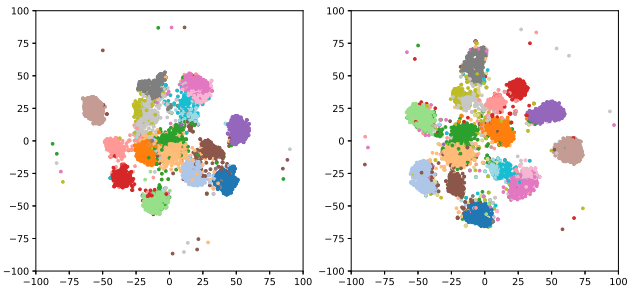
capes. Note that we use the pre-trained model provided by [19] as the inter-domain adaptation model.

Ablation studies. Similar to GTA→Cityscapes, we also validate the effectiveness of each part, *i.e.*, pixel-level adversarial learning (PLA), continuously indexed adversarial learning (CTS), one more round training (T=1) and KLD regularization [52], as shown in Table 7.

Analysis of the proportions of easy pixels. We also conduct an experiment on the proportion of easy pixels for SYNTHIA→Cityscapes. The results are shown in Table 8. It can be observed that our method performs the best when Prop. = 0.66 and our PLA is robust to easy pixel proportions.

7 CONCLUSION

In this paper, we address the shortcoming of image-level intra-domain adaptive segmentation by proposing a pixel-level intra-domain adversarial learning framework. Specifically, we first conduct an inter-domain adaptation and then split the target domain pixels into the easy and hard subdomains. Finally, we propose a pixel-level adversarial learning strategy to conduct the pixel-level intra-domain adaptation from the easy to hard subdomain. Moreover, we propose a continuously indexed adversarial learning technique that can further improve the segmentation accuracy. Experimental results show the effectiveness of our method against existing state-of-the-art approaches.



(a) IntraDA [26] (b) PixIntraDA (ours)

Figure 4: t-SNE [35] visualization of the features extracted by IntraDA [26] and our method on the validation set of GTA→Cityscapes.

8 ACKNOWLEDGEMENTS

The work was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund, and by Guangdong Research Project No. 2017ZT07X152. This work was also supported in part by NSFC-61931024.

REFERENCES

[1] Bowen Cai, Huan Fu, Rongfei Jia, Binqiang Zhao, Hua Li, and Yinghui Xu. 2020. Exploiting Diverse Characteristics and Adversarial Ambivalence for Domain Adaptive Segmentation. *arXiv preprint arXiv:2012.05608* (2020).

[2] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. 2019. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1900–1909.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.

[4] Minghao Chen, Hongyang Xue, and Deng Cai. 2019. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE International Conference on Computer Vision*. 2090–2099.

[5] Minghao Chen, Hongyang Xue, and Deng Cai. 2019. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE International Conference on Computer Vision*. 2090–2099.

[6] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. 2019. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1791–1800.

[7] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. 2017. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*. 1992–2001.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3354–3361.

[10] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. 2019. Dlow: Domain flow for adaptation and generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2477–2486.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*. PMLR, 1989–1998.

[13] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. 2020. Contextual-relation consistent domain adaptation for semantic segmentation. *arXiv preprint arXiv:2007.02424* (2020).

[14] Myeongjin Kim and Hyeran Byun. 2020. Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12975–12984.

[15] Minsu Kim, Sunghun Joung, Seungryong Kim, JungIn Park, Ig-Jae Kim, and Kwanghoon Sohn. 2020. Cross-Domain Grouping and Alignment for Domain Adaptive Semantic Segmentation. *arXiv preprint arXiv:2012.08226* (2020).

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Suhyeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. 2020. Unsupervised Domain Adaptation for Semantic Segmentation by Content Transfer. *arXiv preprint arXiv:2012.12545* (2020).

[18] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. 2020. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision (ECCV)*.

[19] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. 2019. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6936–6945.

[20] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. 2019. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6758–6767.

[21] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. 2017. Predicting deeper into the future of semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 648–657.

[22] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. 2019. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 6778–6787.

[23] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. 2019. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2507–2516.

[24] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. 2020. Instance Adaptive Self-Training for Unsupervised Domain Adaptation. *arXiv preprint arXiv:2008.12197* (2020).

[25] Luigi Musto and Andrea Zinelli. 2020. Semantically Adaptive Image-to-image Translation for Domain Adaptation of Semantic Segmentation. *arXiv preprint arXiv:2009.01166* (2020).

[26] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. 2020. Unsupervised Intra-domain Adaptation for Semantic Segmentation through Self-Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3764–3773.

[27] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2337–2346.

[28] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games. In *European conference on computer vision*. Springer, 102–118.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[30] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3234–3243.

[31] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. 2018. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3752–3761.

[32] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. 2018. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7472–7481.

[33] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. 2019. Domain adaptation for structured output via discriminative patch representations. In *Proceedings of the IEEE International Conference on Computer Vision*. 1456–1465.

[34] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. 2019. Domain adaptation for structured output via discriminative patch representations. In *Proceedings of the IEEE International Conference on Computer Vision*. 1456–1465.

[35] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[36] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. 2019. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2517–2526.

[37] Hao Wang, Hao He, and Dina Katabi. 2020. Continuously Indexed Domain Adaptation. In *ICML*.

[38] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. 2020. Classes Matter: A Fine-grained Adversarial Approach to Cross-domain Semantic Segmentation. *arXiv preprint arXiv:2007.09222* (2020).

[39] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S Huang, and Honghui Shi. 2020. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12635–12644.

[40] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. 2018. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 518–534.

[41] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. 2020. Label-Driven Reconstruction for Domain Adaptation in Semantic Segmentation. *arXiv preprint arXiv:2003.04614* (2020).

[42] Jihan Yang, Ruijia Xu, Ruiyu Li, Xiaojuan Qi, Xiaoyong Shen, Guanbin Li, and Liang Lin. 2020. An Adversarial Perturbation Oriented Domain Adaptation Approach for Semantic Segmentation.. In *AAAI*. 12613–12620.

[43] Yanchao Yang and Stefano Soatto. 2020. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4085–4095.

[44] Fei Yu, Mo Zhang, Hexin Dong, Sheng Hu, Bin Dong, and Li Zhang. 2021. DAST: Unsupervised Domain Adaptation in Semantic Segmentation Based on Discriminator Attention and Self-Training. AAAI.

[45] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. 2019. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *arXiv preprint arXiv:1910.13049* (2019).

[46] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. 2018. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6810–6818.

[47] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Dong Liu, and Tao Mei. 2020. Transferring and Regularizing Prediction for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

9621–9630.

[48] Zhedong Zheng and Yi Yang. 2019. Unsupervised Scene Adaptation with Memory Regularization in vivo. *arXiv preprint arXiv:1912.11164* (2019).

[49] Zhedong Zheng and Yi Yang. 2020. Rectifying Pseudo Label Learning via Uncertainty Estimation for Domain Adaptive Semantic Segmentation. *International Journal of Computer Vision (IJCV)* (2020). https://doi.org/10.1007/s11263-020-01395-y

[50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

[51] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. SEAN: Image Synthesis with Semantic Region-Adaptive Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5104–5113.

[52] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. 2019. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*. 5982–5991.

[53] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*. 289–305.