Joint Adversarial Learning for Domain Adaptation in Semantic Segmentation

Yixin Zhang, Zilei Wang
Department of Automation, University of Science and Technology of China
zhyx12@mail.ustc.edu.cn, zlwang@ustc.edu.cn

Abstract

Unsupervised domain adaptation in semantic segmentation is to exploit the pixel-level annotated samples in the source domain to aid the segmentation of unlabeled samples in the target domain. For such a task, the key point is to learn domain-invariant representations and adversarial learning is usually used, in which the discriminator is to distinguish which domain the input comes from, and the segmentation model targets to deceive the domain discriminator. In this work, we first propose a novel *joint adversarial learning* (JAL) to boost the domain discriminator in output space by introducing the information of domain discriminator from low-level features. Consequently, the training of the high-level decoder would be enhanced. Then we propose a *weight transfer module* (WTM) to alleviate the inherent bias of the trained decoder towards source domain. Specifically, WTM changes the original decoder into a new decoder, which is learned only under the supervision of adversarial loss and thus mainly focuses on reducing domain divergence. The extensive experiments on two widely used benchmarks show that our method can bring considerable performance improvement over different baseline methods, which well demonstrates the effectiveness of our method in the output space adaptation.

Introduction

Recently, deep convolutional neural network (DCNN) has innovated the field of computer vision (Simonyan and Zisserman 2015; Ren et al. 2015; Long, Shelhamer, and Darrell 2015). Its success is largely due to the availability of large-scale and high-quality datasets such as ImageNet (Deng et al. 2009), Pascal VOC (Everingham et al. 2010), COCO (Lin et al. 2014) and Cityscapes (Cordts et al. 2016). Nevertheless, data annotation, especially pixel-level labeling, is labor-intensive and time-consuming, *e.g.*, the average time to annotate a ground-truth image for semantic segmentation in Cityscapes is up to 1 hour (Cordts et al. 2016; Richter et al. 2016). An appealing approach to the issue is to utilize the synthetic data that can be automatically generated and annotated by rendering engine (Richter et al. 2016; Ros et al. 2016). However, the model trained on synthetic data can not generalize well to real-world images, which is caused by the domain shift between the source (*synthetic*) and target (*real-world*) domains (Hoffman et al. 2016). Domain adaptation (Ben-David et al. 2007) is exactly proposed to reduce the domain shift. In this paper, we focus on *unsupervised domain adaptation* in semantic segmentation, where none of image annotation in the target domain is required. A common practice for domain adaptation is to build invariance across domains by minimizing the measure of domain shift such as correlation distances (Ganin and Lempitsky 2014; Tzeng et al. 2017; Long et al. 2016; 2015). In recent works for semantic segmentation, the distribution consistency between the source and target domains is usually enforced via adversarial learning in the pixel space (Hoffman et al. 2018; Wu et al. 2018; Zhang et al. 2018), feature space (Hoffman et al. 2016; Chen et al. 2017; Sankaranarayanan et al. 2018; Zhu et al. 2018) or output space (Tsai et al. 2018; 2019; Luo et al. 2019b). As output space contains richer structured information shared by two domains, it is proven to be more appropriate for semantic segmentation (Tsai et al. 2018). However, how to effectively cooperate adversarial learning to mitigate the domain shift is still an open question.

In this work, we develop a novel framework for domain adaptation in semantic segmentation, which performs adversarial learning in a different way. Figure 1 shows the illustration of our method. Dotted line represents low-level domain discriminator which makes the distribution of two domains closer. Arrow represents *weight transfer module* which can refine the class boundary, thus the segmentation model generalizes well on the target domain.

We analyze the predicted probabilities of domain discriminator in output space adversarial training, and find that for a trained model, the segmentation performance is related to the prediction of domain discriminator which takes segmentation map as input and produces a domain label for each pixel. For a target segmentation map, the more easily the domain discriminator classifies it as source domain, the better performance the segmentation model will achieve. As the segmentation model should deceive domain discriminator, the misclassification indicates the target segmentation map is more similar to source ones. This motivates us to find a more proper way to guide the adversarial training and we propose *joint adversarial learning*. Specifically, the segmen-

tation model should deceive not only the high-level domain discriminator, but also the low-level domain discriminator which is only trained by low-level segmentation maps.

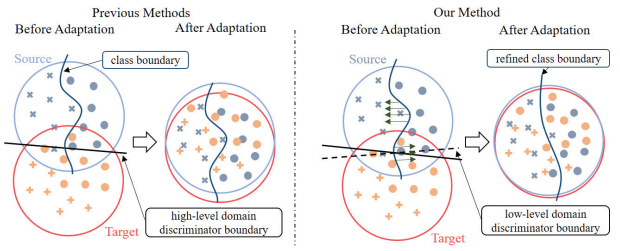

Figure 1: The illustration of the proposed method. The proposed *joint adversarial learning* (dotted line) uses low-level domain discriminator and the *weight transfer module* (arrow) transfers original semantic decoder to a new one that generalizes better in the target domain.

Besides, when introducing *joint adversarial learning*, we find that the decoder is more easily biased towards source domain which makes the model incapable to generalize well in the target domain. That is because the training of decoder is guided by both the segmentation loss and domain classification loss, while the ground truth for the segmentation loss is only from the source (*synthetic*) domain. To tackle the issue, we further proposed a *weight transfer module* to alleviate the bias in the decoder. Specifically, *weight transfer module* takes the weights of decoder as input and assign the output weights to a new decoder which is initialized with identity mapping. This can preserve the knowledge learned on synthetic annotations. Then the decoder with the transferred weights is trained only with the domain adversarial loss. In this case two domains become symmetrical and thus more domain-invariant output can be predicted.

In summary, the main contributions of this work lie in three aspects: (1) we propose *joint adversarial learning* for domain adaptation in semantic segmentation. It utilizes the domain discriminator trained by low-level segmentation maps. In this way, the domain adversarial loss is further boosted. (2) We propose a *weight transfer module* to remove the bias towards the source domain in the decoder. (3) The extensive experiments on GTA5→Cityscapes and SYNTHIA→Cityscapes demonstrate the effectiveness and generalization.

Related Work

Domain adaptation has been a long standing research area. In order to minimize the discrepancy of distributions between source and target domains, some approaches use Maximum Mean Discrepancy (MMD) and its kernel variants (Long et al. 2015; 2016), while others use adversarial approaches (Ganin and Lempitsky 2014; Tzeng et al. 2017).

Hoffman *et al*. (Hoffman et al. 2016) firstly introduce the task of domain adaptation in semantic segmentation by applying adversarial learning on feature representations. Then some work focuses on learning more domain-invariant features by a residual network (Hong et al. 2018) or information bottleneck (Luo et al. 2019a). Instead of directly using internal features, some work (Sankaranarayanan et al. 2018; Zhu et al. 2018) reconstructs the original image from features by an extra generator, and conducts adversarial learning on reconstructed images. There exists some work *et al*. (Saito et al. 2018b; 2018a; Lee et al. 2019) use another perspective of adversarial learning: given two decoders that produce different predictions of the same target image, decoders are trained to maximize the discrepancy, while feature semantic encoder is trained to minimize it. Apart from feature space DA, some methods reduce domain gap in the pixel space (Hoffman et al. 2018; Wu et al. 2018; Zhang et al. 2018) which render the source images with the style of target images and the source labels are still available.

Recently, more approaches perform adversarial learning in output space. Tsai *et al*. (Tsai et al. 2018) first propose to conduct adversarial learning on output space and they use multi-level adaptation to improve the performance. They (Tsai et al. 2019) further propose a classification module on semantic output to produce patch-level representations where adversarial learning is conducted on. In CLAN (Luo et al. 2019b), adversarial force is increased by focusing on category level transferability. Specifically, they assign lower weight for well-aligned output and higher weight for poorly-aligned. In ADVENT (Vu et al. 2019), they consider adversarial learning on entropy map to enforce high prediction certainty on target predictions. Bidirectional Learning (Li, Yuan, and Vasconcelos 2019) achieves state-of-the-art performance by combining image style translation and output space adaptation, two models can be learned alternatively and promote to each other, they also adopt self-supervised learning by using the pseudo target labels to retain the model.

Our work follow the output space adaptation and focus on enhancing the adversarial learning. It can generalize well when using output space adaptation and shows consistent improvements combined with image style translation and pseudo label training. The proposed *weight transfer module* (WTM) is similar to CETL (Chen, Zhang, and Dong 2018). The differences mainly exist in two aspects: 1) They use two different encoders and the decoder is shared. We use shared encoders and different decoders. 2) They use a shared generator to reconstruct the input image and thus transfer knowledge from source encoder to target one. We use the proposed WTM to transfer source knowledge.

Our Approach

In this work, we focus on the unsupervised domain adaptation problem in semantic segmentation. Formally, we are given a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples and a target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled examples. The source domain and target domain are sampled from joint distributions $P(\mathbf{X}_s, \mathbf{Y}_s)$ and $Q(\mathbf{X}_t, \mathbf{Y}_t)$ respectively, and note that $P \neq Q$. Our work aims to learn a segmentation model G that reduces the shifts in the joint distribution across domains and thus generalizes well in target domain. Here we split segmentation model G into semantic encoder Enc and decoder Dec. In the context of adversarial training, domain discriminator is required which

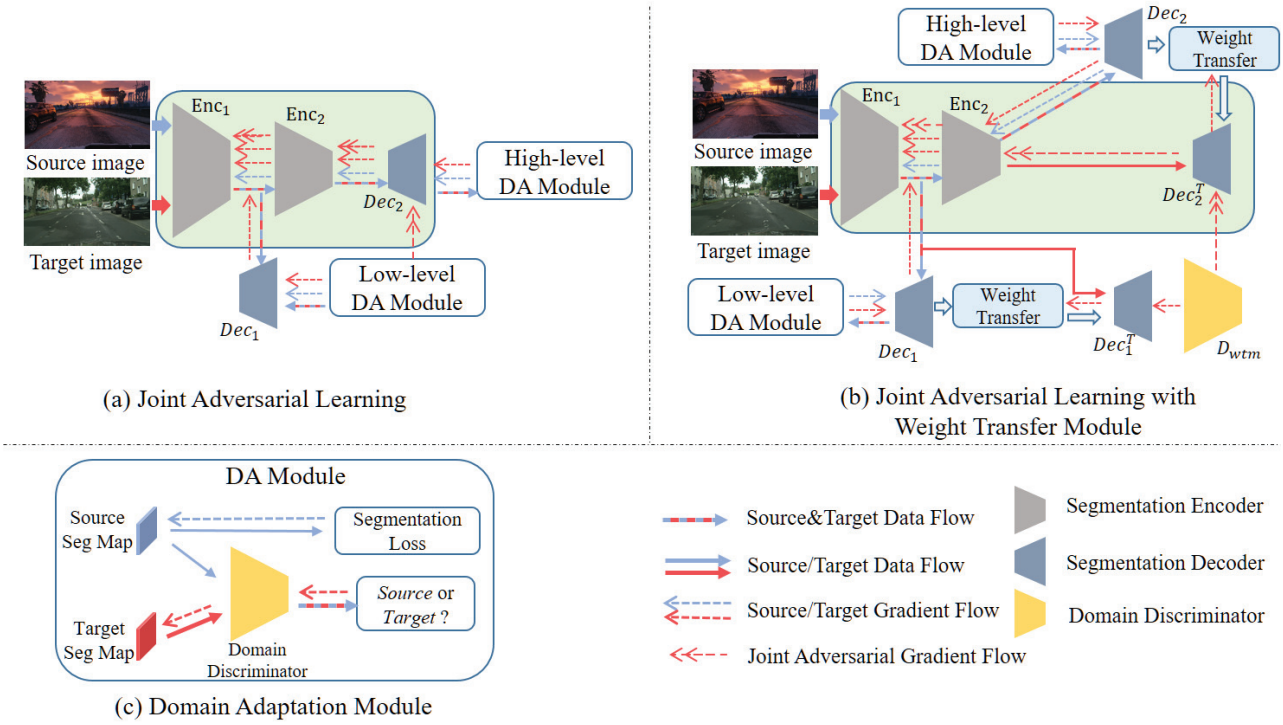

Figure 2: (a) The illustration of proposed *joint adversarial learning*. (b) The framework of our proposed methods combining *joint adversarial learning* and *weight transfer module*. (c) Domain Adaptation Module used after decoder Dec_1 and Dec_2. Note that the models with green background are used for evaluation in the target domain.

is denoted as D. In what follows, we will first review output space adaptation (Tsai et al. 2018) as our background. Then we will dive deeply into the proposed *joint adversarial learning* and *weight transfer module*, and elaborate on how these approaches can improve the adversarial training.

Background

We take AdaptSegNet (Tsai et al. 2018) as our background among different adversarial training methods. The choice is attributed to two reasons: Firstly, the segmentation map contains more shape and structure information, thus the output space adversarial learning is superior to feature level adversarial learning. Secondly, for a given image, different decoders (*e.g.* low-level and high-level decoders) produce similar prediction. As a result, the domain discriminator trained by one set of segmentation maps can be smoothly transferred to another set. This is in accord with the spirit of proposed *joint adversarial learning*.

The main component in AdaptSegNet is Domain Adaptation (DA) Module as shown in figure 2(c), and the framework of AdaptSegNet can be referenced as figure 2(a) without joint adversarial gradient flow.

In DA Module, two losses are involved: segmentation loss and output space domain adversarial loss. The segmentation loss aims at learning discriminative representations with source labeled images:

$$\min_G \mathcal{L}_{seg}(X_s) = -\frac{1}{h \times w} \sum_{h,w} \sum_{c \in C} Y_s \log(P_s^{hwc}), \quad (1)$$

where $P_s = G(X_s) \in \mathbb{R}^{H \times W \times C}$ is the predicted segmentation map (after softmax layer), $H \times W$ is the image size and C is the number of categories.

The adversarial loss acts as a min-max game (Goodfellow et al. 2014), where the training process contains two stages with opposite optimizing objectives. The loss function of training domain discriminator presents as follows:

$$\min_D \mathcal{L}_{dis}(X_s, X_t) = -\frac{1}{h \times w} \sum_{h,w} \log(1 - D(P_t)^{hw}) + \log(D(P_s)^{hw}), \quad (2)$$

$D(P) \in \mathbb{R}^{H \times W}$ is the domain prediction of segmentation map, where 0 indicates the target domain and 1 indicates the source domain.

For the adversarial training of segmentation model, target domain output is used:

$$\min_G \mathcal{L}_{adv}(X_t) = -\frac{1}{h \times w} \sum_{h,w} \log(D(G(X_t))^{hw}). \quad (3)$$

AdaptSegNet use an auxiliary decoder to perform a two-level output space adaptation. Here we use number $i \in$

$\{1, 2\}$ to represent different levels where $i = 1$ means low-level and $i = 2$ means high-level. We further split the encoder Enc into Enc_1 and Enc_2 which denote for low-level and high-level feature encoders. As a result, Dec_1 and Dec_2 are detached after Enc_1 and Enc_2 respectively. Then the overall loss function for segmentation network can be summarized as follows:

$$\min_G \mathcal{L}(X_s, X_t) = \sum_i \lambda^i_{seg}\mathcal{L}^i_{seg}(X_s) + \lambda^i_{adv}\mathcal{L}^i_{adv}(X_t), \tag{4}$$

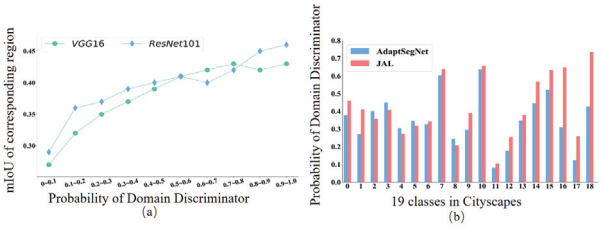

Figure 3: Analysis of domain discriminator predicted probability on the target domain. (a) The tendency of mIoU when the predicted probability increase. Higher probability indicates the corresponding segmentation map being classified as source domain. (b) In the proposed JAL, the mean probabilities where misclassification by domain discriminator are increased.

Joint Adversarial Learning

For better understanding of domain adversarial learning, we analyze the predicted probabilities of domain discriminator. Figure 3(a) indicates the relationship of segmentation performance and discriminator probabilities. It can be observed that segmentation performance is almost positively related to the discriminator prediction. As higher probability indicates the source domain (Eq. 2), we can infer that after training finished, the more the domain discriminator misclassifies, the better performance the segmentation model can achieve. In this way, a better balance between segmentation model and domain discriminator can be reached. This is consistent with the objective of domain adversarial training since the segmentation model should deceive domain discriminator, thus more misclassification in domain discriminator means that the target segmentation map becomes more similar to source ones.

This motivates us to find a discriminator that can achieve a better balance between segmentation model and domain discriminator. In this case, the segmentation model can deceive the discriminator more successfully. We have tested discriminator with more parameters or multiple discriminators, but the improvements are marginal. Finally, we focus on segmentation model itself and propose a novel *joint adversarial learning* (JAL). As shown in Figure 2(a), the segmentation model should deceive not only the high-level domain discriminator, but also the low-level domain discriminator. It is worth noting that low-level domain discriminator is only trained by low-level segmentation maps, and the high-level maps are not involved. Specifically, only one loss is added:

$$\min_G \mathcal{L}_{adv\_jal}(X_t) = -\frac{1}{h \times w}\sum_{h,w} \log(D_1(G(X_t))), \tag{5}$$

where D_1 denotes the domain discriminator in low-level DA Module. $G(X_t)$ represents $Dec_2(Enc_2(Enc_1(X_t)))$.

Figure 3(b) shows the class-wise probabilities produced by different domain discriminators. It can be found that low-level domain discriminator tends to classify the target segmentation map as the source domain. The experiment is based on benchmark GTA5→ Cityscapes with VGG16 backbone, and JAL shows great improvement over baseline (36.7% *vs.* 35.0%).

Although JAL could improve the performance of VGG16 backbone, it behaves differently with ResNet101. As shown in figure 4, JAL outperforms AdaptSegNet (Tsai et al. 2018) at first, but as training prolonging, JAL gradually performs worse than AdaptSegNet. This phenomenon is similar to overfitting in AdaptSegNet when the training iteration increases. The difference is that we add an adversarial loss and the overfitting comes earlier.

In AdaptSegNet, the overfitting phenomenon is caused by two aspects: Firstly, the segmentation model is supervised by source label and no target ground truth is provided, thus the model will capture more specific details in the source domain. In the context of output space adaptation, these details mean the differences of shape (*e.g.* the shape of traffic sign), spatial layout and even the label distribution (*e.g.* the class train is rarer in GTA5 than in Cityscapes). Secondly, domain adversarial learning reduces domain gap by pushing the distribution of segmentation maps in the target domain to those in the source domain. As a result, training more iteration makes the model bias towards source domain and hinder generalization in the target domain. In the proposed JAL, the auxiliary adversarial loss aggravates the overfitting phenomenon and ResNet101 which is more powerful makes the overfitting severer.

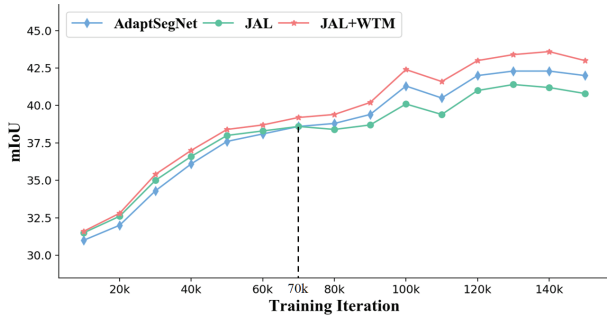

Figure 4: We show the tendency of mIoU on the target domain by different methods. After $70k$ iteration, JAL starts to performs worse than AdaptSegNet. Adding the proposed WTM can address this problem. The experiments are conducted with the same random seed.

Weight Transfer Module

From the above analysis, the key to addressing the overfitting phenomenon is alleviating the bias towards the source domain. AdaptSegNet uses early stopping, but in JAL, if we stop at turning point (iteration $70k$ in figure 4), the model is not fully trained. We argue that when adopting shared weights for both segmentation loss and domain adversarial loss, it is hard to control the effect of source segmentation loss without harming the whole training. As our goal is to remove bias towards the source domain, we propose to build a new decoder that is not directly affected by source segmentation loss. We achieve this by *weight transfer module* (WTM). Specifically, WTM transfers the original decoder to a new one and the new decoder is only trained by adversarial loss. It is also important for the new decoder to be discriminative among different classes, thus we initialize *weight transfer module* as unit mapping. During training, the transferred decoder keeps similar with the original decoder and the knowledge learned from the source domain can be transferred.

As single adversarial loss is not a strong constraint, we choose the transfer function to be simple and easy to learn. In the implementation, we use a single convolution layer as WTM, and we add it on both low-level and high-level decoders as shown in Figure 2(b). The weights of new decoders can be formulated as follows:

$$W_{Dec_1^T} = WTM_1 * W_{Dec_1}, \quad (6)$$
$$W_{Dec_2^T} = WTM_2 * W_{Dec_2},$$

where $*$ represents convolution operation, WTM_1 is low-level *weight transfer module*. W_{Dec_1} and $W_{Dec_1^T}$ represent the weights of the original and transferred low-level decoder respectively. $WTM_2, W_{Dec_2}, W_{Dec_2^T}$ are denoted in a similar way. The transferred decoder takes the same features as the original decoder, and produces transferred segmentation maps.

$$P_s^{1T} = Dec_1^T(Enc_1(X_s)), P_s^{2T} = Dec_2^T(Enc(X_s)), \quad (7)$$
$$P_t^{1T} = Dec_1^T(Enc_1(X_t)), P_t^{2T} = Dec_2^T(Enc(X_t)),$$

where $Enc(X)$ represents $Enc_2(Enc_1(X))$.

As the new decoders are only trained by adversarial loss, we use the third domain discriminator D_{wtm}. For the training of D_{wtm}, the segmentation map from transferred low-level decoder is used:

$$\min_{D_{wtm}} \mathcal{L}_{dis\_wtm}(X_s, X_t) = -\frac{1}{h \times w} \sum_{h,w} \quad (8)$$
$$[\log(D_{wtm}(P_s^{1T})^{hw}) + \log(1 - D_{wtm}(P_t^{1T})^{hw})],$$

D_{wtm} is used for both low-level and high-level adversarial learning:

$$\min_{G,WTM_i} \mathcal{L}^i_{adv\_wtm}(X_t) = -\frac{1}{hw} \sum_{h,w} \log(D_{wtm}(P_t^{iT})^{hw}), \quad (9)$$

where $i = \{1, 2\}$. When $i = 2$, the domain discriminator D_{wtm} is also used to train high-level decoder thus the *joint adversarial learning* is introduced.

It is worth noting that once the training finished, we first transfer decoder Dec_2 to Dec_2^T, then semantic encoder Enc and decoder Dec_2^T are used together for testing.

The overall loss function for segmentation model can be formulated as follows:

$$\min_{G,WTM_i} \mathcal{L}(X_s, X_t) = \sum_i \lambda^i_{seg} \mathcal{L}^i_{seg}(X_s) + \quad (10)$$
$$\lambda^i_{adv} \mathcal{L}^i_{adv}(X_t) + \lambda^i_{adv\_wtm} \mathcal{L}^i_{adv\_wtm}(X_t),$$

where $i = \{1, 2\}$, and $\lambda^i_{seg}, \lambda^i_{adv}, \lambda^i_{adv\_wtm}$ are used to balance different losses.

Experiments

Experimental Details

Datasets. We use the popular *synthetic-2-real* domain adaptation set-ups, *e.g.*, GTA5→Cityscapes and SYNTHIA→Cityscapes. Cityscapes (Cordts et al. 2016) is a real-world dataset which contains urban street images collected from a moving vehicle captured in 50 cities around Germany and neighboring countries. Training set of 2975 images is involved in the training phase. GTA5 (Richter et al. 2016) consists of $24,966$ synthesized frames rendered by the gaming engine GTAV. Here 19 classes are adopted for training and evaluation. SYNTHIA (Ros et al. 2016) contains with $9,400$ synthesized images. Inheriting from existing methods (Zhang, David, and Gong 2017; Wu et al. 2018; Zhu et al. 2018; Zou et al. 2018), we train our models with 16 classes and evaluating on 16- and 13-class subsets. In both set-ups, 500 images of Cityscapes validation set are employed to evaluation.

Network architectures. We utilize the DeepLab-v2 (Chen et al. 2018) framework as semantic segmentation model. Following (Chen et al. 2018), we modify the stride and dilation rate of the last two convolution blocks. After the final convolution layer, we use the Atrous Spatial Pyramid Pooling (ASPP) (Chen et al. 2018) as the final decoder. We experiment on two different backbones: VGG16 (Simonyan and Zisserman 2015) and ResNet101 (He et al. 2016). For low-level adaptation, we add ASPP decoder on *conv5* of VGG16 and *conv4* of ResNet101 (choice of position of low-level decoder can be referenced in supplementary material).

For different domain discriminators used in this work, we adopt the same network architecture. Specifically, we use a similar structure with DCGAN (Radford, Metz, and Chintala 2016), which consists of 5 convolution layers with kernel 4×4 with channel numbers {64,128,256,512,1} and stride of 2. Each convolution layer is followed by a Leak-ReLU (Maas, Hannun, and Ng 2013) with the slope of -0.2.

In WTM, we transfer original ASPP decoder (Chen et al. 2018) to a new one. The ASPP decoder contains four convolution kernels with different dilation rates. Each one has the shape of $1024 \times 19 \times 3 \times 3$. Here we conduct transformation process in a convolutional way where the original weights are regarded as input map. The kernel size of transfer function is set to 1. As a result, *weight transfer module* is a convolutional layer with kernel size $1024 \times 1024 \times 1 \times 1$.

Implementation details. All images are resized and cropped to 1024×512. Segmentation model is trained by

Table 1: Results of adapting GTA5 to Cityscapes. "V" and "R" denotes backbone of VGG16 and ResNet101.

Method	Backbone	mIoU
AdaptSegNet (Tsai et al. 2018)	V	35.0
Patch Adv (Tsai et al. 2019)	V	37.5
CLAN (Luo et al. 2019b)	V	36.1
ADVENT (Vu et al. 2019)	V	35.6
Source Only	V	30.5
Ours(Dec_2^T)	V	**38.5**
AdaptSegNet (Tsai et al. 2018)	R	42.4
Patch Adv (Tsai et al. 2019)	R	43.2
CLAN (Luo et al. 2019b)	R	43.2
ADVENT (Vu et al. 2019)	R	42.7
Source Only	R	36.8
Ours(Dec_2^T)	R	**43.5**

Table 2: Results of adapting SYNTHIA to Cityscapes. "V" and "R" denotes backbone of VGG16 and ResNet101. mIoU and mIoU* represents performance of 16- and 13-class respectively.

Method	Backbone	mIoU	mIoU*
AdaptSegNet (Tsai et al. 2018)	V	-	37.6
Patch Adv (Tsai et al. 2019)	V	33.7	39.6
CLAN (Luo et al. 2019b)	V	-	39.3
ADVENT (Vu et al. 2019)	V	31.4	36.6
Source Only	V	28.0	32.4
Ours(Dec_2^T)	V	**36.2**	**42.2**
AdaptSegNet (Tsai et al. 2018)	R	-	46.7
Patch Adv (Tsai et al. 2019)	R	40.0	46.5
CLAN (Luo et al. 2019b)	R	-	47.8
ADVENT (Vu et al. 2019)	R	40.8	47.6
Source Only	R	33.9	38.9
Ours(Dec_2^T)	R	**41.6**	**48.3**

SGD optimizer with learning rate 0.00025, momentum 0.9 and weight decay 0.0001. Domain discriminators are trained by Adam optimizer with learning rate 0.0001. As for *weight transfer module*, we use SGD optimizer with learning rate 0.0001, momentum 0.9 and weight decay 0. We use the polynominal annealing procedure in (Chen et al. 2018) to schedule the learning rate. The max iteration is 250k, and we use early stopping at 150k. For hyper-parameters in Eq. (10), We set λ^1_{adv}=0.0002, λ^2_{adv}=0.001, λ^1_{seg}=0.1, and λ^2_{seg}=1.0 following AdaptSegNet. We also set $\lambda^1_{adv\_wtm} = 0.0002$ and $\lambda^2_{adv\_wtm} = 0.001$.

Overall Results

GTA5→Cityscapes: We report semantic segmentation performance in Table 1. We report the mIoU of decoder Dec_2^T as our final results.

With VGG16 backbone, our method outperforms other methods. With ResNet101 backbone, our method achieves slightly better result compared other methods (Tsai et al. 2019; Luo et al. 2019b) based on output space adaptation. For Patch Adv, we report the result of ResNet101 without pixel-level adaptation and pseudo label training. We also show the combination of these methods and proposed components in ablation study where a stronger baseline is used. For ADVENT (Vu et al. 2019) which conducts adaptation based on entropy map, we report the performance of their adversarial learning method which also adopts a two-level adaptation. It can be seen that compared with replacing the softmax prediction output with entropy map, out method which resorts to low-level output can bring more improvement. Figure 5 shows the visualization of domain discriminator output for target domain images. It can be seen that more target domain regions are classified as source domain, and segmentation results of corresponding regions are improved.

SYNTHIA→Cityscapes: Table 2 shows results on the 16- and 13-class subsets of the Cityscapes validation set. Our method shows great superiority compared with other methods of both VGG16 and ResNet101 backbones. As images in SYNTHIA cover more diverse viewpoints than the ones in GTA5 and Cityscapes, ADVENT (Vu et al. 2019) uses class-ratio prior to prevent the model to get biased towards some easy classes for VGG16 backbone. Our method outperforms ADVENT by $+4.8\%$ indicating that it is more stable for different synthetic scenes.

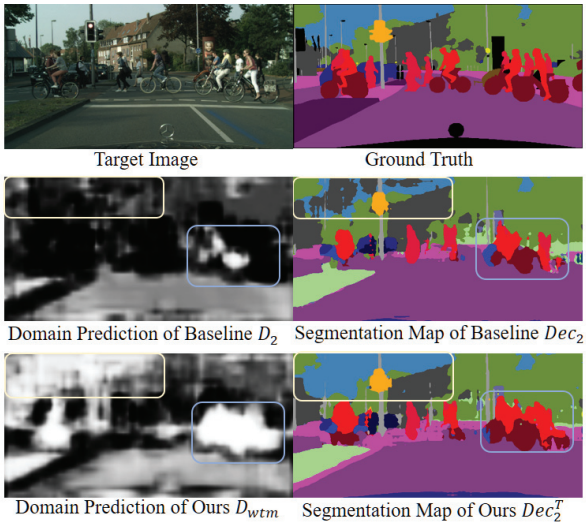

Figure 5: Visualization of segmentation maps and corresponding domain discriminator predictions. For the probabilities produced by domain discriminator, we scale them to [0,255]. In accordance with Eq 2, brightness (higher probability) indicates the input segmentation map belongs to the source domain. Here the benchmark GTA5→Cityscapes with ResNet101 is used. The image is from Cityscapes validation set.

Ablation study

In this section, we perform several exploratory studies to give more insight into the functionality and effectiveness of the proposed approach.

Different Methods for Enhancing Adversarial Learning

We also try other methods for enhancing adversarial learning. The result is given in Table 3. Baseline represents the domain discriminator described before. For ASPP discriminator, we add an ASPP module before the baseline discriminator. To get a wider discriminator, we increase the number of channels in each layer, the width denotes for the multiplication coefficient of channel number. We also compare the proposed JAL with multiple discriminators which are trained by high-level segmentation maps. Although these methods could improve the performance, they are limited when introduce more parameters or discriminators. The proposed JAL can largely improve the performance which indicates the effectiveness of boosting the adversarial learning.

Table 3: Different methods for enhancing adversarial learning. The experiments are conducted on GTA5→Cityscapes benchmark with VGG16 backbone.

Methods	Settings	mIoU
Baseline	width=1	35.0
ASPP Discriminator	−−	35.3
Multi Discriminator	N=2	35.4
	N=3	35.5
	N=4	35.5
Wider Discriminator	width=2	35.6
	width=4	35.6
JAL	−−	36.7

The Effect of Each Component

In this section, we want to explore how each component affects adversarial learning. Table 4 shows the ablation study results of *joint adversarial learning* (JAL) and *weight transfer module* (WTM). The first row represents the baseline model which adopts two-level output space domain adaptation. The second row corresponds to the situation adding JAL. With VGG16 backbone, JAL brings $+1.7\%$ gain over baseline. With ResNet101 backbone, JAL does not perform well due to the *overfitting phenomenon* as described before.

Row $4-6$ shows the effect of WTM. For VGG16 and ResNet101 backbones, WTM shows consistent improvements compared when added to both high-level and low-level decoders. Although the improvement brought by WTM is relatively small, it can greatly benefit the adversarial learning combined with JAL. These two modules are complementary and WTM is necessary in case of deeper network like ResNet101.

Combination of Stronger Baseline

To validate the generalization of the proposed method, we choose another stronger baseline Bidirectional Learning (Li, Yuan, and Vasconcelos 2019) which achieves the state of art performance. Bidirectional Learning uses CycleGAN (Zhu et al. 2017) to render the source images to target style. The adapted segmentation model works as a perceptual loss to further promote the image style translation. The image translation model and domain adaptation model are trained iteratively and the max iteration number is set to 2.

Table 4: Component wise ablation studies corresponding to the GTA5 → Cityscapes setting.

	JAL	**WTM** on Dec_1	**WTM** on Dec_2	**mIoU** VGG16	**mIoU** Res101
1				35.0	42.4
2	√			36.7	41.5
3		√		35.9	42.6
4			√	36.1	42.7
5		√	√	36.4	42.9
6	√	√	√	**38.5**	**43.5**

We combined our method with Bidirectional Learning in two different settings as shown in Table 5. ST represents image style translation. SSL denotes for self-supervised learning which is added in domain adaptation step. It can be found that our method can bring consistent improvements over different baselines and is complementary to pixel-level adaptation and pseudo target training.

Table 5: Combination of Bidirectional Learning and our methods on GTA5 → Cityscapes setting. We adopt the backbone of ResNet101.

Settings	Iteration	Baseline	+Ours	Δ
ST	1	42.7	43.8	1.1
	2	43.3	44.2	0.9
ST+SSL	1	47.2	48.0	0.8
	2	48.5	49.3	0.8

Conclusion

In this paper, we propose a novel method named *joint adversarial learning* for domain adaptation in semantic segmentation. It uses the low-level domain discriminator to provide auxiliary adversarial loss for the training high-level decoder. Besides, we propose *weight transfer module* to remove the bias towards the source domain. It transfers the original decoder to a new decoder which is only trained by adversarial loss, thus the two domains become symmetrical and more domain-invariant output can be predicted. Extensive experimental results on two widely used benchmarks validate the effectiveness and generalization of our method.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant 61673362 and 61836008, Youth Innovation Promotion Association CAS (2017496).

References

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. In *NIPS*.

Chen, Y.-H.; Chen, W.-Y.; Chen, Y.-T.; Tsai, B.-C.; Frank Wang, Y.-C.; and Sun, M. 2017. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*.

Chen, S.; Zhang, C.; and Dong, M. 2018. Coupled end-to-end transfer learning with generalized fisher information. In *CVPR*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*.

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

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*.

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

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hoffman, J.; Wang, D.; Yu, F.; and Darrell, T. 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A. A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. *ICML*.

Hong, W.; Wang, Z.; Yang, M.; and Yuan, J. 2018. Conditional generative adversarial network for structured domain adaptation. In *CVPR*.

Lee, C.-Y.; Batra, T.; Baig, M. H.; and Ulbricht, D. 2019. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*.

Li, Y.; Yuan, L.; and Vasconcelos, N. 2019. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. *ICML*.

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

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Luo, Y.; Liu, P.; Guan, T.; Yu, J.; and Yang, Y. 2019a. Significance-aware information bottleneck for domain adaptive semantic segmentation. *arXiv preprint arXiv:1904.00876*.

Luo, Y.; Zheng, L.; Guan, T.; Yu, J.; and Yang, Y. 2019b. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*.

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

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for data: Ground truth from computer games. In *ECCV*. Springer.

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A. M. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*.

Saito, K.; Ushiku, Y.; Harada, T.; and Saenko, K. 2018a. Adversarial dropout regularization. *ICLR*.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018b. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*.

Sankaranarayanan, S.; Balaji, Y.; Jain, A.; Nam Lim, S.; and Chellappa, R. 2018. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*.

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

Tsai, Y.-H.; Hung, W.-C.; Schulter, S.; Sohn, K.; Yang, M.-H.; and Chandraker, M. 2018. Learning to adapt structured output space for semantic segmentation. In *CVPR*.

Tsai, Y.-H.; Sohn, K.; Schulter, S.; and Chandraker, M. 2019. Domain adaptation for structured output via discriminative representations. In *ICCV*.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*.

Vu, T.-H.; Jain, H.; Bucher, M.; Cord, M.; and Pérez, P. 2019. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*.

Wu, Z.; Han, X.; Lin, Y.-L.; Gokhan Uzunbas, M.; Goldstein, T.; Nam Lim, S.; and Davis, L. S. 2018. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *ECCV*.

Zhang, Y.; Qiu, Z.; Yao, T.; Liu, D.; and Mei, T. 2018. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*.

Zhang, Y.; David, P.; and Gong, B. 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*.

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

Zhu, X.; Zhou, H.; Yang, C.; Shi, J.; and Lin, D. 2018. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *ECCV*.

Zou, Y.; Yu, Z.; Vijaya Kumar, B.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*.