DecoupleNet: Decoupled Network for Domain Adaptive Semantic Segmentation

Xin Lai[1], Zhuotao Tian[1], Xiaogang Xu[1], Yingcong Chen[3,4,5]⋆, Shu Liu[2], Hengshuang Zhao[6,7], Liwei Wang[1], and Jiaya Jia[1,2]

[1]CUHK [2]SmartMore [3]HKUST(GZ) [4]HKUST
[5]HKUST(GZ)-SmartMore Joint Lab [6]HKU [7]MIT

Abstract. Unsupervised domain adaptation in semantic segmentation alleviates the reliance on expensive pixel-wise annotation. It uses a labeled source domain dataset as well as unlabeled target domain images to learn a segmentation network. In this paper, we observe two main issues of existing domain-invariant learning framework. (1) Being distracted by the feature distribution alignment, the network cannot focus on the segmentation task. (2) Fitting source domain data well would compromise the target domain performance. To address these issues, we propose DecoupleNet to alleviate source domain overfitting and let the final model focus more on the segmentation task. Also, we put forward Self-Discrimination (SD) and introduce an auxiliary classifier to learn more discriminative target domain features with pseudo labels. Finally, we propose Online Enhanced Self-Training (OEST) to contextually enhance the quality of pseudo labels in an online manner. Experiments show our method outperforms existing state-of-the-art methods. Extensive ablation studies verify the effectiveness of each component. Code is available at `https://github.com/dvlab-research/DecoupleNet`.

Keywords: Unsupervised Domain Adaptation · Semantic Segmentation

1 Introduction

Semantic segmentation has made tremendous progress in recent years and it has benefited plenty of applications. Its performance highly relies on pixel-wise annotation. In this paper, we alleviate data-reliance and focus on unsupervised domain adaptation (UDA). We learn a segmentation network with a labeled source-domain dataset (usually a physically synthetic dataset) and an unlabeled target domain dataset.

Due to "domain shift" [13,62] between the source and target domains, directly adopting the model trained on the source domain causes performance degradation on the target one. To minimize domain shift, domain-invariant learning [64,65,69,46,14,70] aligns distributions of source and target features. Specifically, the features or predictions from different domains are aligned with

⋆ Corresponding Author

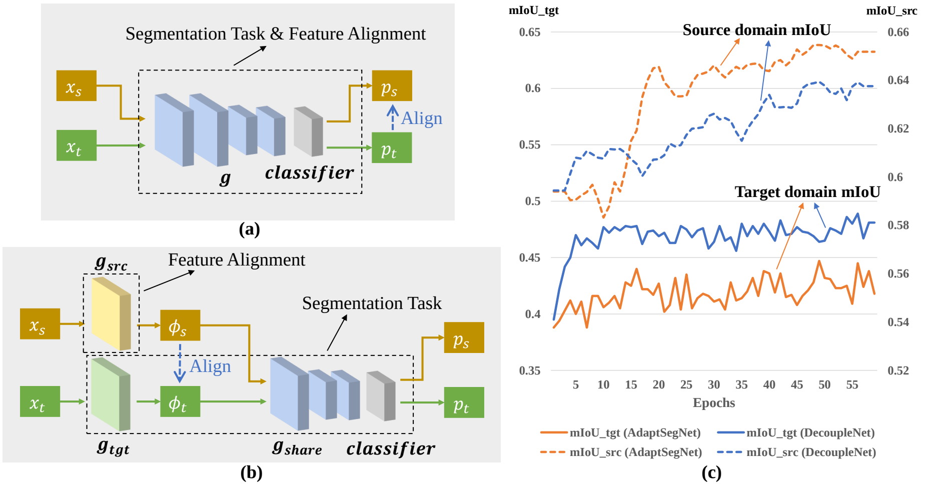


Fig. 1. (a) Domain-invariant learning. (b) Our proposed DecoupleNet. The original encoder $\boldsymbol{g}$ is split into $\boldsymbol{g}_{tgt}$ and $\boldsymbol{g}_{share}$. Also, $\boldsymbol{g}_{src}$ and $\boldsymbol{g}_{tgt}$ share the same architecture but not the parameters. The source-domain shallow features ϕ_s are aligned towards ϕ_t with an adversarial loss. During inference, $\boldsymbol{g} = \boldsymbol{g}_{share} \circ \boldsymbol{g}_{tgt}$ is used, and $\boldsymbol{g}_{src}$ is simply discarded. (c) Plot of the validation mIoU on source domain data (mIoU_src) by the dashed line and on target domain data (mIoU_tgt) by the solid line. We compare our DecoupleNet (blue line) with a representative domain-invariant learning method, i.e., AdaptSegNet [64] (orange line)

a discriminator by adversarial learning, as shown in Fig. 1(a). The discriminator learns to distinguish between source and target features, while the segmentation network learns to generate features that can fool the discriminator.

Domain-invariant learning alleviates domain shift. However, we still observe the following two problems.

(1) *Tasks entanglement.* The feature distribution alignment and the segmentation task are conducted simultaneously in a single network, as shown in Fig. 1(a). Being distracted by feature distribution alignment, the network cannot focus on semantic segmentation, leading to inferior performance.

(2) *Source domain overfitting.* Since the training objective involves cross-entropy loss that minimizes errors on the source domain data, the trained model would fit the source domain data well, as shown in Fig. 1(c). However, in UDA, we only care about the performance on the target domain, regardless of how it performs on the source domain. Moreover, as we will discuss in Sec. 3.2, fitting the source domain very well would contrarily compromise the target domain performance.

Based on these two observations, we design DecoupleNet to decouple feature distribution alignment and the segmentation task. As shown in Fig. 1(b), we introduce a copy of shallow encoder layers for the source domain, i.e., $\boldsymbol{g}_{src}$, during training. Our goal is to let $\boldsymbol{g}_{src}$ conduct feature distribution alignment, such that the final model $\boldsymbol{g} = \boldsymbol{g}_{share} \circ \boldsymbol{g}_{tgt}$ focuses more on the downstream segmentation task. Also, it is notable that $\boldsymbol{g}_{src}$ is simply discarded during inference, and it only incurs negligible computational costs during training as shown in the supplementary material.

With our new design, the issue of *tasks entanglement* can be addressed, as shown in Fig. 1(b). Moreover, during training, we only require the model $\boldsymbol{g}_{share} \circ \boldsymbol{g}_{src}$ to fit well on the source domain, but never require the final model $\boldsymbol{g} = \boldsymbol{g}_{share} \circ \boldsymbol{g}_{tgt}$ to do so. Thus, the final model avoids overfitting in the source domain. As shown in Fig. 1(c), compared to the domain-invariant method (AdaptSegNet [64]), DecoupleNet alleviates the *source domain overfitting* problem, and boosts the target domain performance.

In addition, in order to learn more discriminative features for the target domain, we propose the Self-Discrimination (SD) technique by virtue of pseudo labels. Unlike most self-training-based methods [90,59,51,83,77,29,70], SD does not need another training phase to re-train the whole network from scratch. Instead, pseudo labels are generated at each training iteration and can be employed as an additional supervision in an online manner. Given the fact that directly adopting the noisy pseudo labels to supervise itself could corrupt the existing classifier, we introduce an auxiliary classifier during training to prevent contamination.

Finally, we propose Online Enhanced Self-Training (OEST) to further boost the performance by extending DecoupleNet to a multi-stage training paradigm. Most existing self-training-based methods [90,59,51,77,29,70] directly use the generated pseudo labels without updating them in the re-training process. Contrarily, at each training iteration, OEST updates the pseudo labels by fusing current contextually enhanced predictions, which effectively improves the quality of pseudo labels.

In summary, our contribution is threefold.

- We propose DecoupleNet to decouple feature distribution alignment and semantic segmentation. This lets the network avoid tasks entanglement and focus more on the segmentation task.
- To learn more discriminative features, we put forward Self-Discrimination by introducing an auxiliary classifier. Moreover, we propose Online Enhanced Self-Training to contextually enhance the quality of pseudo labels.
- Experiments show that our approach outperforms existing state-of-the-art methods by a large margin. Also, extensive ablation studies verify the effectiveness of each component in our method.

2 Related Work

Semantic segmentation. Semantic segmentation aims to assign a class label to every pixel in an image. FCN [58] is a classic semantic segmentation network, which introduces a fully-convolutional network. Considering that the final output size of FCN is smaller than the input, methods based on encoder-decoder structures [52,2,56] are proposed to refine the output. Though the high-level feature has already encoded the semantic information, it cannot well capture the long-range relationship. Dilated convolution [5,81], global pooling [38], pyramid pooling [87,86,79] and attention mechanism [15,26,88,89] are used to better incorporate the context. Despite the success, all the models need annotations to accomplish training, which costs much human effort.

Unsupervised domain adaptation. Unsupervised dmain adaptation [19] intends to alleviate the data-reliance with a labeled dataset from a different domain. Distance-based methods [42,43,44,67,61,33,72,41] minimize the distribution distance such as MMD [67] between the source and target domain. With the development of Generative Adversarial Network (GAN) [18], adversarial learning methods [16,66,74,24,37,75,9,12,36,68,30,47,85,3,28,1,40] get popular to align the marginal or conditional feature distributions between the source and target domains. Also, methods of [11,54] factorize the feature into domain-specific and domain-agnostic features.

UDA in Semantic Segmentation. AdaptSegNet [64] employs adversarial learning to align predictions between the source and target domain in the output space and method of [45] makes further improvement. Patch-level information is used in [65] to improve the performance and contextual relationship is considered in [25,27] explicitly. In [84,71,53], feature distance is directly minimized. In [69,76,32,63,78,39,31,82,60], semi-supervised learning methods, such as entropy minimization, adding perturbation, contrastive learning and randomly dropout, further boost performance. Methods of [70,14,46] align class-conditioned feature distribution. Those of [34,50,48,73] provide distinct processing for features from different domains on some modules. On the other hand, image-to-image translation methods were considered in [23,35,17,77,8,80]. Recently, self-training-based methods [90,59,51,83,77,29,70,21,20] re-train the network with the pseudo labels generated from the initial network, yielding considerable improvement.

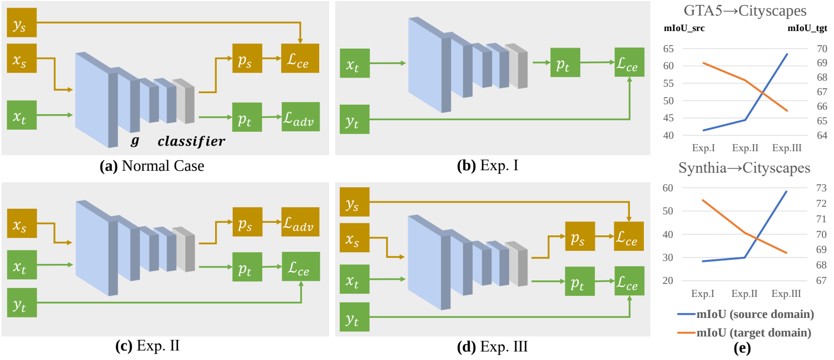
3 Our Method

In this section, we first introduce the preliminary in Section 3.1. Then, the key observations are presented as our motivation in Sec. 3.2. Afterwards, DecoupleNet, SD and OEST are elaborated in Sections 3.3, 3.4 and 3.5, respectively.

3.1 Preliminary

Problem definition. We define the source domain images $\mathcal{X}_s$ along with ground-truth labels $\mathcal{Y}_s$, and the unlabeled target domain images $\mathcal{X}_t$. Our goal is to train a segmentation model $\mathcal{G}$ that performs well on the target domain.

A representative domain-invariant solution [64] is shown in Fig. 2 (a). The source and target domain images (x_s, x_t) pass forward the segmentation network



(a) Normal Case

(b) Exp. I

(c) Exp. II

(d) Exp. III

(e)

Fig. 2. (a) A representative domain-invariant method, AdaptSegNet [64]. The normal case: the ground-truth labels for the target domain are not available. The brown and green lines represent the source and target domain branches, respectively. (b) Exp. I: training by target domain images and their ground-truth labels with CE loss. (c) Exp. II: training by source domain images with the adversarial loss, as well as the target domain images and labels with CE loss. The discriminator is not shown in the figure. (d) Exp. III: training by both source and target domains images and labels with two CE losses. Note that unlike the normal case in (a), we use target domain ground-truth labels in the toy experiments to support our idea only rather than give a complete solution. (e) Evaluation results of two benchmarks on both source (blue line) and target (orange line) domain validation sets (best viewed in color)

$\mathcal{G}$, which is typically composed of an encoder $\boldsymbol{g}$ and a classifier $\mathcal{C}$, to obtain the predictions (p_s, p_t), respectively. It is written as

$$p_s = \mathcal{C}(\boldsymbol{g}(x_s)), \quad p_t = \mathcal{C}(\boldsymbol{g}(x_t)). \tag{1}$$

For the source domain prediction p_s, the cross-entropy loss is employed with its ground-truth label y_s as

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbf{1}\{y_{s,i} = c\}\log p_{s,i,c}, \tag{2}$$

where N is the number of spatial locations in the source prediction map p_s, C is the number of classes, $y_{s,i}$ represents the class label at the i-th location, and $p_{s,i,c}$ represents the source prediction score of the c-th class at the i-th location.

As for the target domain prediction p_t, a discriminator $\mathcal{D}$ is used to align the distributions of the source and target predictions. The adversarial loss $\mathcal{L}_{adv}$ is defined as

$$\mathcal{L}_{adv} = \frac{1}{N_d}\sum_{i=1}^{N_d}(\mathcal{D}(p_t)_i - 0)^2, \tag{3}$$

where N_d is the number of spatial locations in the discriminator output, 0 is the label of the source domain, and we follow LSGAN [49] to use the MSE Loss.

The final loss $\mathcal{L}_{seg}$ for the segmentation network is defined as

$$\mathcal{L}_{seg} = \mathcal{L}_{ce} + \lambda_{adv}\mathcal{L}_{adv}, \tag{4}$$

where λ_{adv} controls the weight for $\mathcal{L}_{adv}$. To train the discriminator, the discriminator loss $\mathcal{L}_d$ is defined as

$$\mathcal{L}_d = \frac{1}{N_d}\sum_{i=1}^{N_d}(\mathcal{D}(p_s)_i - 0)^2 + \frac{1}{N_d}\sum_{i=1}^{N_d}(\mathcal{D}(p_t)_i - 1)^2, \tag{5}$$

where the labels of source and target domain are 0 and 1, respectively. Training alternates between updating the segmentation network with $\mathcal{L}_{seg}$ and the discriminator with $\mathcal{L}_d$.

3.2 Motivation

The method above aligns the distributions of source and target domain features for domain-invariant learning. However, as shown in Fig. 2(a), since the learning objective involves $\mathcal{L}_{ce}$ during training, the trained network has to fit the source domain data very well. The source domain overfitting issue potentially impairs the segmentation performance on the target domain.

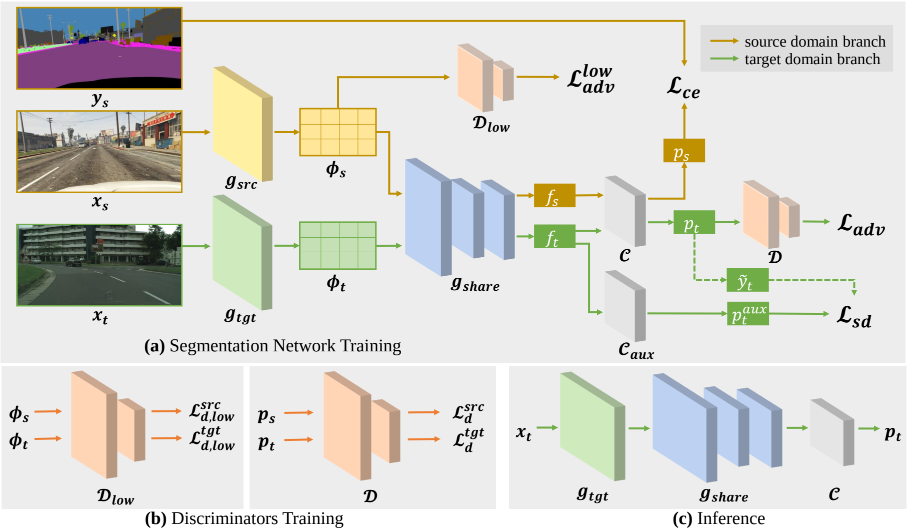
We conduct three experiments to verify this fact, and show them in Fig. 2(b)-(d). Unlike the normal case (Fig. 2(a)), we use the target domain ground-truth labels in the toy experiments only to support our idea rather than give a solution. As shown in Fig. 2(e), from Exp. I to II, we apply an extra adversarial loss, so the model performs slightly better on the source domain data. Further, from Exp. II to III, we apply a stronger CE loss on the source domain, so it performs very well on the source domain. However, the results in Fig. 2(e) reveal the fact that the better the model fits on the source domain data, the worse it performs on the target domain. This exactly supports our idea, i.e., overfitting the source domain data actually impairs the final performance on the target domain.

Motivated by the observations, we propose a new framework to decouple the feature distribution alignment from the segmentation task. It alleviates the issue of source domain overfitting, and enables the final model to focus more on target-domain semantic segmentation.

3.3 DecoupleNet

The framework of DecoupleNet is shown in Fig. 3. We first split the feature encoder $\boldsymbol{g}$ into two parts, i.e., $\boldsymbol{g}_{tgt}$ and $\boldsymbol{g}_{share}$. Besides, we maintain another module $\boldsymbol{g}_{src}$, which shares the same architecture with $\boldsymbol{g}_{tgt}$. The source and target domain images (x_s, x_t) are fed into the source blocks $\boldsymbol{g}_{src}$ and target blocks $\boldsymbol{g}_{tgt}$ to yield the shallow features (ϕ_s, ϕ_t), respectively. They further pass through the shared blocks $\boldsymbol{g}_{share}$ to get the features (f_s, f_t). Afterwards, they are passed into the classifier $\mathcal{C}$ to obtain the predictions (p_s, p_t). Initially, we have

$$\phi_s = \boldsymbol{g}_{src}(x_s), \quad f_s = \boldsymbol{g}_{shared}(\phi_s), \quad p_s = \mathcal{C}(f_s), \tag{6}$$



(a) Segmentation Network Training

(b) Discriminators Training

(c) Inference

Fig. 3. Framework of DecoupleNet. (a) Segmentation network training. The brown line represents the source branch, while the green line denotes the target branch. Note that the dashed line means stopping gradients. (b) Discriminators training. (c) Inference pipeline. Best viewed in color

$$\phi_t = \boldsymbol{g}_{tgt}(x_t), \quad f_t = \boldsymbol{g}_{shared}(\phi_t), \quad p_t = \mathcal{C}(f_t). \tag{7}$$

Then, we adopt cross-entropy loss $\mathcal{L}_{ce}$ for the labeled source domain data as

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbf{1}\{y_{s,i} = c\}\log p_{s,i,c}. \tag{8}$$

Besides, we require the distribution of the source-domain shallow features ϕ_s to align towards that of the target domain, i.e., ϕ_t, since our goal is to let the source blocks $\boldsymbol{g}_{src}$ bear the responsibility of feature distribution alignment. Specifically, adversarial learning is adopted for the shallow feature alignment with an additional discriminator $\mathcal{D}_{low}$ and an adversarial loss $\mathcal{L}_{adv}^{low}$ as

$$\mathcal{L}_{adv}^{low} = \frac{1}{N_d^{low}}\sum_{i=1}^{N_d^{low}}(\mathcal{D}_{low}(\phi_s)_i - 1)^2, \tag{9}$$

where N_d^{low} denotes the number of locations in the discriminator output, and 1 is the label of the target domain.

The design of DecoupleNet is with the following considerations. Basically, the source domain images differ from the target ones mainly on low-level information, such as illumination and texture. Also, it is known that the shallow layers in a network often do well in capturing the low-level information. With these facts, it is natural to let the source blocks g_{src} align the source-domain shallow features towards the target ones.

Practically, the shallow feature distribution alignment by $\mathcal{L}_{adv}^{low}$ may be imperfect, and the shallow features for the source and target domains may still be slightly mismatched. To remedy them, we use the adversarial loss $\mathcal{L}_{adv}$ in the output space, as defined in Eq. (3). In this way, we have the final loss $\mathcal{L}_{seg}$ for training the segmentation network defined as

$$\mathcal{L}_{seg} = \mathcal{L}_{ce} + \lambda_{adv}^{low}\mathcal{L}_{adv}^{low} + \lambda_{adv}\mathcal{L}_{adv}, \tag{10}$$

where λ_{adv}^{low} and λ_{adv} control the contributions of the corresponding loss. It is notable that the incorporation of $\mathcal{L}_{adv}$ only brings minor improvement (+0.3% mIoU), as shown in Exp. 5 and 6 of Table 3. This shows the feature alignment is mainly attributed to $\mathcal{L}_{adv}^{low}$. The $\mathcal{L}_{adv}$ only serves as a complement.

To train the discriminators, as shown in Fig. 3(b), we follow previous work [64] to yield the discriminator loss as

$$\mathcal{L}_{d}^{low} = \frac{1}{N_d^{low}}\sum_{i=1}^{N_d^{low}}(\mathcal{D}_{low}(\phi_s)_i - 0)^2 + \frac{1}{N_d^{low}}\sum_{i=1}^{N_d^{low}}(\mathcal{D}_{low}(\phi_t)_i - 1)^2, \tag{11}$$

$$\mathcal{L}_{d} = \frac{1}{N_d}\sum_{i=1}^{N_d}(\mathcal{D}(p_s)_i - 0)^2 + \frac{1}{N_d}\sum_{i=1}^{N_d}(\mathcal{D}(p_t)_i - 1)^2. \tag{12}$$

During inference, as shown in Fig. 3(c), we adopt $\mathcal{F} = \mathcal{C} \circ g_{share} \circ g_{tgt}$ as the final model. All other modules are simply discarded. Note that we do not introduce extra parameters during inference.

Advantage of DecouleNet. First, the source blocks $\boldsymbol{g}_{src}$ now bear the responsibility of feature distribution alignment. Being less distracted by feature alignment, the final model (i.e., $\boldsymbol{g}_{share} \circ \boldsymbol{g}_{tgt}$) focuses more on the segmentation task. Second, though the source domain branch $\boldsymbol{g}_{share} \circ \boldsymbol{g}_{src}$ needs to directly fit the source domain data with $\mathcal{L}_{ce}$, the final model $\boldsymbol{g} = \boldsymbol{g}_{share} \circ \boldsymbol{g}_{tgt}$ is never required to perform well on the source domain during training. This alleviates the source domain overfitting problem and facilitates performance boosting on the target domain, as shown in Fig. 1(c).

3.4 Self-Discrimination

Despite the effectiveness of DecoupleNet, the target blocks $\boldsymbol{g}_{tgt}$ is updated only according to $\mathcal{L}_{adv}$, which may not be strong enough to learn optimal parameters for $\boldsymbol{g}^{tgt}$. Also, without a proper learning objective for the target domain, the features f_t may not be discriminative enough. To address them, we propose Self-Discrimination (SD) to provide more supervision on the target domain branch.

As shown in Fig. 3(a), we introduce an auxiliary classifier $\mathcal{C}_{aux}$, which shares the same architecture with the main classifier $\mathcal{C}$. As the target domain feature f_t passes the classifier $\mathcal{C}$ to yield p_t, we also forward f_t into the auxiliary classifier

$\mathcal{C}^{aux}$ to get the auxiliary prediction p_t^{aux}. Meanwhile, we calculate the pseudo label $\tilde{y}_t$ according to the main prediction p_t. Similar to [90], we adopt class-wise thresholds $\boldsymbol{\tau}$ to ignore uncertain pixels in the pseudo labels and maintain class balancing as well. Finally, we yield the self-discrimination loss $\mathcal{L}_{sd}$ as

$$p_t^{aux} = \mathcal{C}_{aux}(f_t), \quad \hat{y}_{t,i} = \operatorname*{argmax}_{c=1}^{C} p_{t,i,c}, \quad \tilde{y}_{t,i} = \begin{cases} \hat{y}_{t,i} & p_{t,i,c=\hat{y}_{t,i}} \geq \boldsymbol{\tau}_{c=\hat{y}_{t,i}} \\ -1(ignored) & otherwise \end{cases},$$

$$\mathcal{L}_{sd} = -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{c=1}^{C} \mathbf{1}\{\tilde{y}_{t,i} = c\} \log p_{t,i,c}^{aux}, \tag{13}$$

where C is the number of classes, N_t is the number of spatial locations in the auxiliary prediction map p_t^{aux}, $\boldsymbol{\tau}_c$ is the threshold for the c-th class, $p_{t,i,c}$ and $p_{t,i,c}^{aux}$ are the main and auxiliary prediction scores of the c-th class for the feature at the i-th location, respectively, and $\tilde{y}_{t,i}$ is the pseudo label for the i-th location in the prediction map p_t.

It is notable that the class-wise thresholds $\boldsymbol{\tau}$ are initialized to zero when starting training. It is updated with the current predictions p_t at each iteration. The implementation details are given in the supplementary material.

Basically, $\mathcal{L}_{sd}$ is a cross-entropy loss applied to $p_{t,i,c}^{aux}$. It has a nice property that can adaptively scale the gradients with the current prediction error. Hence, it is capable of yielding more discriminative target features f_t. To verify the effectiveness, we compare the t-SNE visualizations with and without SD in the supplementary material. During inference, we only use the main classifier, and the auxiliary classifier is simply discarded.

Remarkably, the accuracy of the pseudo labels is more than 80%, and continues to increase during training, as shown in Fig. 4. Therefore, although there might be wrong supervision from pseudo labels, the benefits brought by SD still outweigh the risks.

Finally, we incorporate $\mathcal{L}_{sd}$ into the final segmentation loss $\mathcal{L}_{seg}$ as

$$\mathcal{L}_{seg} = \mathcal{L}_{ce} + \lambda_{adv}^{low} \mathcal{L}_{adv}^{low} + \lambda_{adv} \mathcal{L}_{adv} + \lambda_{sd} \mathcal{L}_{sd}.$$

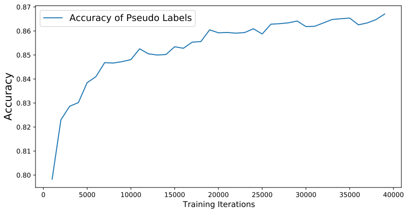


Fig. 4. Accuracy of pseudo labels during training

The auxiliary classifier. The auxiliary classifier plays an important role in SD. If we directly apply the self-discrimination loss $\mathcal{L}_{sd}$ on the main prediction p_t without the auxiliary classifier, the noisy pseudo labels may corrupt the normal training of the main classifier with $\mathcal{L}_{ce}$ and cause large performance degradation, as shown in Exp. 1 and 2 of Table 6. In contrast, introducing an auxiliary classifier avoids the side effect on the main classifier.

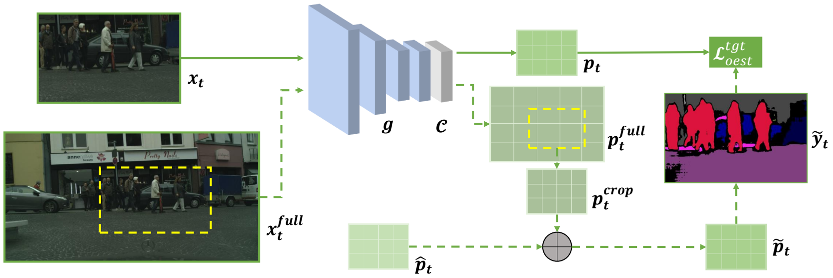


Fig. 5. Framework of Online Enhanced Self-Training. Dashed line: stopping gradients

3.5 Online Enhanced Self-Training

To further boost performance, we extend DecoupleNet from a single stage to a multi-stage self-training paradigm. Most existing self-training-based methods [90,59,51,83,77,29,70] generate pseudo labels in the re-labeling phase and directly use them to provide supervision without further update in the re-training phase. Generally, the predictions get more accurate during the re-training process. Fixing the generated pseudo labels may lead to inferior performance. ProDA [83] uses prototypes to denoise the pseudo labels. But it requires to maintain an extra momentum encoder and needs to update the prototypes at each iteration. In contrast, we propose a simple yet effective method, i.e., Online Enhanced Self-Training (OEST), to contextually enhance the pseudo labels via a simple average operation at each iteration.

The framework of OEST is given in Fig. 5. After the first-stage training explained in Sections 3.3 and 3.4, we generate the pseudo soft labels $\hat{p}_t \in [0,1]^{H\times W\times C}$ by making predictions on each target domain training image x_t using the trained model. Then, in the re-training process, we pass the target domain image crops x_t with strong data augmentation (e.g., color jitter) into the segmentation network $\mathcal{G} = \mathcal{C} \circ \boldsymbol{g}$ to yield their predictions p_t. In addition, we forward their corresponding full images x_t^{full} with weak data augmentation (e.g., random horizontal flip) to obtain the full predictions p_t^{full} as

$$p_t = softmax(\mathcal{G}(x_t)), \quad p_t^{full} = softmax(\mathcal{G}(x_t^{full})). \tag{14}$$

Afterwards, we crop p_t^{crop} from p_t^{full} in the same way as cropping x_t from x_t^{full}, and enhance the original pseudo soft labels $\hat{p}_t$ with p_t^{crop} via a simple average operation to yield $\tilde{p}_t$. It follows by ignoring uncertain pixels with class-wise thresholds $\boldsymbol{\tau}^{st}$ as in [90] to obtain the updated pseudo labels $\tilde{y}_t$ as

$$\tilde{p}_t = \frac{1}{2}(\hat{p}_t + p_t^{crop}), \quad \hat{y}_{t,i} = \operatorname*{argmax}_{c=1}^{C} \tilde{p}_{t,i,c}, \quad \tilde{y}_{t,i} = \begin{cases} \hat{y}_{t,i} & \tilde{p}_{t,i,c=\hat{y}_{t,i}} \geq \boldsymbol{\tau}^{st}_{c=\hat{y}_{t,i}} \\ -1 (ignored) & otherwise \end{cases}.$$

Since p_t^{crop} is aware of the contexts in the full image, the quality of the original pseudo labels can be contextually enhanced via simple fusion. Finally, we yield

Table 1. Results on GTA5→Cityscapes with ResNet101 and DeepLabv2. ST: self-training

Method	ST	*road*	*sw.*	*build*	*wall*	*fence*	*pole*	*light*	*sign*	*veg.*	*terrain*	*sky*	*person*	*rider*	*car*	*truck*	*bus*	*train*	*moto.*	*bicycle*	mIoU
SourceOnly		27.0	20.6	53.9	20.8	19.4	35.3	40.7	23.0	84.6	30.1	73.5	63.9	31.4	65.7	10.5	26.3	2.1	34.1	21.8	36.0
AdaptSeg [64]		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
AdaptSeg*(LS)*		**91.4**	**48.4**	81.2	27.4	21.2	31.2	35.3	16.1	84.1	32.5	78.2	57.7	28.2	85.9	33.8	43.5	0.2	23.9	16.9	44.1
CLAN [46]		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 [69]		89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	**36.7**	78.8	58.7	30.5	84.8	**38.5**	**44.5**	1.7	31.6	32.4	45.5
FADA [70]		88.5	39.7	**83.6**	**37.9**	24.7	27.5	34.1	21.3	83.3	32.9	**83.4**	58.0	33.5	84.7	37.9	39.8	25.2	30.8	27.6	47.1
Ours		87.5	37.6	83.2	31.6	**28.3**	**38.6**	**44.3**	**24.9**	**85.1**	31.0	76.0	**68.1**	**36.9**	**86.4**	28.4	39.0	**25.5**	**42.8**	**36.1**	**49.0**
CBST [90]	✓	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
AdaptPatch [65]	✓	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
Label-Driven [77]	✓	90.8	41.4	84.7	35.1	27.5	31.2	38.0	32.8	85.6	42.1	84.9	59.6	34.4	85.0	**42.8**	52.7	3.4	30.9	38.1	49.5
FADA [70]	✓	91.0	50.6	86.0	**43.4**	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
Kim et al. [29]	✓	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-MBT [80]	✓	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
TPLD [59]	✓	**94.2**	**60.5**	82.8	36.6	16.6	39.3	29.0	25.5	85.6	**44.9**	84.4	60.6	27.4	84.1	37.0	47.0	31.2	36.1	50.3	51.2
IAST [51]	✓	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
MetaCorrection [21]	✓	92.8	58.1	86.2	39.7	33.1	36.3	42.0	38.6	85.5	37.8	87.6	62.8	31.7	84.8	35.7	50.3	2.0	36.8	48.0	52.1
DPL [8]	✓	92.8	54.4	86.2	41.6	32.7	36.4	49.0	34.0	85.8	41.3	86.0	63.2	34.2	87.2	39.3	44.5	18.7	42.6	43.1	53.3
ProDA [83]	✓	91.5	52.3	82.9	41.8	35.7	40.3	44.3	**43.2**	87.1	43.4	79.6	66.6	31.6	86.9	40.1	53.0	0.0	45.7	53.2	53.6
Ours+ST	✓	88.5	47.8	**87.4**	38.3	**36.9**	**44.9**	**53.8**	39.6	**88.0**	38.7	**88.8**	**70.4**	**39.4**	**87.8**	31.4	**55.0**	**37.4**	**47.1**	**55.9**	**56.7**
ProDA *(w/ SimCLR)*	✓	**87.8**	**56.0**	79.7	**46.3**	**44.8**	45.6	53.5	**53.5**	88.6	**45.2**	82.1	70.7	39.2	**88.8**	**45.5**	**59.4**	1.0	48.9	56.4	57.5
Ours *(w/ SimCLR)*	✓	87.6	49.3	**87.2**	42.5	41.6	**46.6**	**57.4**	44.0	**89.0**	43.9	**90.6**	**73.0**	**43.8**	88.1	32.9	53.7	**44.3**	**49.8**	**57.2**	**59.1**

the self-training loss $\mathcal{L}_{oest}^{tgt}$ on p_t with $\tilde{y}_t$, and add it to the source domain CE loss $\mathcal{L}_{ce}^{src}$ to obtain the final loss $\mathcal{L}$ as

$$\mathcal{L}_{oest}^{tgt} = -\frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{c=1}^{C}\mathbf{1}\{\tilde{y}_{t,i}=c\}\log p_{t,i,c}, \quad \mathcal{L} = \mathcal{L}_{ce}^{src} + \mathcal{L}_{oest}^{tgt}. \tag{15}$$

4 Experiment

4.1 Implementation Details

Experimental setting. Following previous work [64,70,46,69,45,65,14], we use the ResNet-101 [22] and DeepLabv2 [4] as our base model. To split the feature encoder, we take {layer0, layer1} as the target blocks $\boldsymbol{g}_{tgt}$ and the rest as the shared blocks $\boldsymbol{g}_{share}$ for GTA5 dataset, while {layer0, layer1, layer2} as $\boldsymbol{g}_{tgt}$ and the rest as $\boldsymbol{g}_{share}$ for Synthia dataset. Note that layer0 refers to {conv1, bn1, relu, maxpool}. More details are given in the supplementary material.

Datasets. Following most previous work, evaluation is performed on GTA5 → Cityscapes, Synthia → Cityscapes and Cityscapes → Cross-City. The details of the datasets [55,57,10,7] are given in the supplementary material.

4.2 Results

The comparison with existing state-of-the-art methods is given in Tables 1 and 2. Clearly, our method outperforms others by a large margin. Previous meth-

Table 2. Results on Synthia→Cityscapes with ResNet101 and DeepLabv2. ST: self-training. mIoU^{+}: mIoU of 13 classes

Method	ST	*road*	*sw.*	*build*	*wall*	*fence*	*pole*	*light*	*sign*	*veg.*	*sky*	*person*	*rider*	*car*	*bus*	*moto.*	*bicycle*	mIoU	mIoU^{+}
SourceOnly		59.9	24.7	57.7	6.3	0.0	32.5	**29.7**	15.0	72.8	70.8	59.2	17.7	73.0	23.0	11.6	22.6	36.0	41.4
AdaptSeg [64]		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
AdaptSeg*(LS)*		84.0	40.5	79.3	10.4	0.2	22.7	6.5	8.0	78.3	82.7	56.3	22.4	74.0	33.2	18.9	**34.9**	40.8	47.6
CLAN [46]		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 [69]		**85.6**	**42.2**	79.7	8.7	**0.4**	25.9	5.4	8.1	80.4	**84.1**	57.9	23.8	73.3	**36.4**	14.2	33.0	41.2	48.0
Ours		77.9	38.9	74.4	**11.9**	0.2	**33.3**	26.5	**17.1**	**83.6**	80.0	**60.7**	**26.5**	**79.9**	26.4	**25.5**	33.5	**43.5**	**50.1**
CBST [90]	✓	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	42.6	48.9
AdaptPatch [65]	✓	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
FADA [70]	✓	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	45.2	52.5
Label-Driven [77]	✓	85.1	44.5	81.0	-	-	-	16.4	15.2	80.1	84.8	59.4	31.9	73.2	41.0	32.6	44.7	-	53.1
Kim et al. [29]	✓	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
FDA-MBT [80]	✓	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
TPLD [59]	✓	80.9	44.3	82.2	19.9	0.3	40.6	20.5	30.1	77.2	80.9	60.6	25.5	84.8	41.1	24.7	43.7	47.3	53.5
MetaCorrection [21]	✓	**92.6**	**52.7**	81.3	8.9	2.4	28.1	13.0	7.3	83.5	85.0	60.1	19.7	84.8	37.2	21.5	43.9	45.1	52.5
DPL [8]	✓	87.5	45.7	82.8	13.3	0.6	33.2	22.0	20.1	83.1	86.0	56.6	21.9	83.1	40.3	29.8	45.7	47.0	54.2
IAST [51]	✓	81.9	41.5	**83.3**	17.7	**4.6**	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	33.1	52.7	49.8	57.0
ProDA [83]	✓	87.3	44.2	**83.3**	26.6	0.3	41.8	43.8	**33.1**	86.7	82.4	69.1	25.7	88.0	50.3	31.1	43.8	52.3	59.1
Ours+ST	✓	78.7	47.4	75.7	**27.8**	1.0	**43.3**	**49.1**	32.6	**87.8**	**87.3**	**69.3**	**34.4**	**88.5**	**55.0**	**44.8**	**58.5**	**55.1**	**62.2**
ProDA *(w/ SimCLR)*	✓	**87.8**	45.7	**84.6**	**37.1**	0.6	44.0	**54.6**	37.0	**88.1**	84.4	**74.2**	24.3	88.2	51.1	40.5	45.6	55.5	62.0
Ours *(w/ SimCLR)*	✓	77.8	**48.6**	75.6	32.0	**1.9**	**44.4**	52.9	**38.5**	87.8	**88.1**	71.1	**34.3**	**88.7**	**58.8**	**50.2**	**61.4**	**57.0**	**64.1**

ods [64,65,69,70] neglect the adverse effect brought by entanglement of feature distribution alignment and the segmentation task. Contrarily, DecoupleNet decouples these two tasks, and boosts the performance.

Further, equipped with OEST, our method demonstrates stronger performance. It is also notable that our method even surpasses ProDA [83] by 3.1 points on GTA5→Cityscapes and 2.8 points on Synthia→Cityscapes, achieving a new state-of-the-art result. It is notable that following ProDA to distill the SimCLR [6] initialized student, our method still outperforms ProDA on both benchmarks. On Cityscapes → Cross-City, our method also manifests competitive results given in the supplementary material.

4.3 Ablation Study

DecoupleNet. Comparing Exp. 2 and 4 in Table 3 reveals that DecoupleNet outperforms the domain-invariant method (AdaptSegNet [64]) by 3.0% mIoU, which reveals the effectiveness of DecoupleNet. Note that except the decoupled architecture and $\mathcal{L}_{adv}^{low}$, Exp. 2 and 4 are kept all the same for fair comparison.

Notably, we emphasize that $\mathcal{L}_{adv}$ brings only slight improvement (+0.3% mIoU) by comparing Exp. 6 and 7 in Table 3. On the other hand, $\mathcal{L}_{adv}^{low}$ brings large performance boost (+2.2% mIoU), in the comparison between Exp. 5 and 7. This shows that the huge performance boost by DecoupleNet mainly comes from the decoupled network architecture and $\mathcal{L}_{adv}^{low}$, rather than $\mathcal{L}_{adv}$. $\mathcal{L}_{adv}$ only serves as a complement for the imperfect alignment by $\mathcal{L}_{adv}^{low}$. This demonstrates the effectiveness of DecoupleNet from another perspective.

Table 3. Ablation study for DecoupleNet and SD. Decouple: decoupled network architecture. SD: Self-Discrimination

ID	Method	Decouple	$\mathcal{L}_{ce}$	$\mathcal{L}_{adv}$	$\mathcal{L}_{adv}^{low}$	$\mathcal{L}_{sd}$	mIoU
1	SourceOnly		✓				36.0
2	AdaptSegNet		✓	✓			44.1
3	AdaptSegNet + SD		✓	✓		✓	46.0
4	DecoupleNet	✓	✓	✓	✓		47.1
5	DecoupleNet + SD (*w/o* $\mathcal{L}_{adv}^{low}$)	✓	✓	✓		✓	46.8
6	DecoupleNet + SD (*w/o* $\mathcal{L}_{adv}$)	✓	✓		✓	✓	48.7
7	DecoupleNet + SD	✓	✓	✓	✓	✓	**49.0**
8	DecoupleNet (*w/o* $\mathcal{L}_{adv}^{low}$)	✓	✓	✓			44.7

Table 4. Ablation study for the decoupled layers, i.e., the architecture of $\boldsymbol{g}_{src}$ or $\boldsymbol{g}_{tgt}$. Note that ResNet has 5 layers in total, and layer0 refers to the stem layer, i.e., {conv1, bn1, relu, maxpool}

Decoupled layers	{layer0}	{layer0,1}	{layer0,1,2}
mIoU (%)	47.7	**49.0**	47.9

Besides, we investigate the effect of decoupled layers (i.e., the architecture of $\boldsymbol{g}_{src}$ or $\boldsymbol{g}_{tgt}$) in Table 4. Making it too shallow leads to insufficient capability for feature alignment, while making it too deep may interfere segmentation.

Table 5. Ablation study for the alignment direction between ϕ_s and ϕ_t. $\phi_s \rightarrow \phi_t$: applying $\mathcal{L}_{adv}^{low}$ on ϕ_s. $\phi_t \rightarrow \phi_s$: applying $\mathcal{L}_{adv}^{low}$ on ϕ_t. No alignment: $\lambda_{adv}^{low} = 0$

Alignment direction	$\phi_s \rightarrow \phi_t$	$\phi_t \rightarrow \phi_s$	No alignment
mIoU (%)	**49.0**	47.3	46.8

Also, we highlight the importance of the alignment direction of ϕ_s and ϕ_t in Table 5. $\phi_s \rightarrow \phi_t$ performs the best. We explain that this prevents the segmentation network $\boldsymbol{g} = \boldsymbol{g}_{share} \circ \boldsymbol{g}_{tgt}$ from being distracted by feature alignment.

Self-Discrimination. By comparing Exp. 4 and 7 in Table 3, we observe a performance boost of 1.9% mIoU brought by SD, which clearly demonstrates its effectiveness. Also, it is notable that when we directly apply SD on the domain-invaraint method (i.e., AdaptSegNet [64]), the performance still continues to improve by a large margin, through the comparison between Exp. 2 and 3 in Table 3. It shows that SD is not limited to DecoupleNet and can serve as a plugin to existing methods by providing an additional supervision.

In addition, we show the t-SNE visualizations of the target domain features f_t with and without SD in the supplementary material. It reveals the fact that the model tends to learn more discriminative target domain features with SD.

Moreover, to show the necessity of the auxiliary classifier, we make comparison in Table 6. For the model w/o auxiliary classifier (Exp. 2), we directly apply $\mathcal{L}_{sd}$ on the main predictions p_t, which leads to large degradation (-3.0% mIoU) compared to Exp. 1. We conjecture that the supervision signal from the noisy pseudo labels may interfere the normal training of the main classifier with source domain ground-truth labels. Further, Exp. 1 and 3 in Table 6 show the effectiveness of the class-wise thresholds, since it alleviates the class-imbalance issue on pseudo labels.

Table 6. Ablation study for class-wise thresholds and the auxiliary classifier. class-balance: class-wise thresholds. aux: auxiliary classifier

ID	class-balance	aux	mIoU	Δ
1	✓	✓	49.0	0.0
2	✓		46.0	-3.0
3		✓	48.4	-0.6

Table 7. Ablation study for OEST. avg *(full)*: average pseudo soft labels and predictions from full images. avg *(crop)*: average pseudo soft labels and predictions from crops. fix: use fixed pseudo soft labels only. pred only: use predictions only

Fusion Method	avg *(full)*	avg *(crop)*	fix	pred only
mIoU (%)	**56.7**	55.2	55.6	25.8

Online Enhanced Self-Training. As shown in Table 7, we compare the models with various fusion methods. The comparison between 'avg *(full)*' and 'avg *(crop)*' show the effectiveness of contextual enhancement via full predictions. Moreover, 'fix' is inferior to 'avg *(full)*' by 1.1% mIoU, which shows that online updating pseudo labels with current predictions indeed improves the quality of pseudo labels and brings performance boost. As for 'pred only', it totally corrupts the training potentially due to the instability of the online prediction.

5 Conclusion

We have observed two issues of existing domain-invariant learning methods – *tasks entanglement* and *source domain overfitting*. We propose DecoupleNet to enable the final model to focus more on the segmentation task. Moreover, Self-Discrimination is put forward to learn more discriminative target features. Finally, we design OEST to contextually enhance the pseudo labels.

References

1. Awais, M., Zhou, F., Xu, H., Hong, L., Luo, P., Bae, S.H., Li, Z.: Adversarial robustness for unsupervised domain adaptation. In: ICCV (2021)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. TPAMI (2017)
3. Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J.: Progressive feature alignment for unsupervised domain adaptation. In: CVPR (2019)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2017)
5. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2018)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
7. Chen, Y.H., Chen, W.Y., Chen, Y.T., Tsai, B.C., Frank Wang, Y.C., Sun, M.: No more discrimination: Cross city adaptation of road scene segmenters. In: ICCV (2017)
8. Cheng, Y., Wei, F., Bao, J., Chen, D., Wen, F., Zhang, W.: Dual path learning for domain adaptation of semantic segmentation. In: ICCV (2021)
9. Cicek, S., Soatto, S.: Unsupervised domain adaptation via regularized conditional alignment. In: ICCV (2019)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
11. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: CVPR (2020)
12. Deng, Z., Luo, Y., Zhu, J.: Cluster alignment with a teacher for unsupervised domain adaptation. In: ICCV (2019)
13. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML (2014)
14. Du, L., Tan, J., Yang, H., Feng, J., Xue, X., Zheng, Q., Ye, X., Zhang, X.: Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In: ICCV (2019)
15. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: CVPR (2019)
16. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)
17. Gong, R., Li, W., Chen, Y., Gool, L.V.: Dlow: Domain flow for adaptation and generalization. In: CVPR (2019)
18. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. arXiv:1406.2661 (2014)
19. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV (2011)
20. Guizilini, V., Li, J., Ambruș, R., Gaidon, A.: Geometric unsupervised domain adaptation for semantic segmentation. In: ICCV (2021)

21. Guo, X., Yang, C., Li, B., Yuan, Y.: Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation. In: CVPR (2021)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
23. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
24. Hong, W., Wang, Z., Yang, M., Yuan, J.: Conditional generative adversarial network for structured domain adaptation. In: CVPR (2018)
25. Huang, J., Lu, S., Guan, D., Zhang, X.: Contextual-relation consistent domain adaptation for semantic segmentation. In: ECCV (2020)
26. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: ICCV (2019)
27. Kang, G., Wei, Y., Yang, Y., Zhuang, Y., Hauptmann, A.: Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. NeurIPS (2020)
28. Kang, G., Zheng, L., Yan, Y., Yang, Y.: Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In: ECCV (2018)
29. Kim, M., Byun, H.: Learning texture invariant representation for domain adaptation of semantic segmentation. In: CVPR (2020)
30. Kurmi, V.K., Kumar, S., Namboodiri, V.P.: Attending to discriminative certainty for domain adaptation. In: CVPR (2019)
31. Lai, X., Tian, Z., Jiang, L., Liu, S., Zhao, H., Wang, L., Jia, J.: Semi-supervised semantic segmentation with directional context-aware consistency. In: CVPR (2021)
32. Lee, S., Kim, D., Kim, N., Jeong, S.G.: Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In: ICCV (2019)
33. Li, S., Xie, M., Lv, F., Liu, C.H., Liang, J., Qin, C., Li, W.: Semantic concentration for domain adaptation. In: ICCV (2021)
34. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. arXiv:1603.04779 (2016)
35. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: CVPR (2019)
36. Liu, H., Long, M., Wang, J., Jordan, M.: Transferable adversarial training: A general approach to adapting deep classifiers. In: ICML (2019)
37. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. arXiv:1606.07536 (2016)
38. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv (2015)
39. Liu, W., Ferstl, D., Schulter, S., Zebedin, L., Fua, P., Leistner, C.: Domain adaptation for semantic segmentation via patch-wise contrastive learning. arXiv:2104.11056 (2021)
40. Liu, X., Guo, Z., Li, S., Xing, F., You, J., Kuo, C.C.J., El Fakhri, G., Woo, J.: Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate. In: ICCV (2021)
41. Liu, X., Li, S., Ge, Y., Ye, P., You, J., Lu, J.: Recursively conditional gaussian for ordinal unsupervised domain adaptation. In: ICCV (2021)
42. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML (2015)
43. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. arXiv:1602.04433 (2016)

44. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML (2017)
45. Luo, Y., Liu, P., Guan, T., Yu, J., Yang, Y.: Significance-aware information bottleneck for domain adaptive semantic segmentation. In: ICCV (2019)
46. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: CVPR (2019)
47. Ma, X., Zhang, T., Xu, C.: Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In: CVPR (2019)
48. Mancini, M., Porzi, L., Bulo, S.R., Caputo, B., Ricci, E.: Boosting domain adaptation by discovering latent domains. In: CVPR (2018)
49. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: ICCV (2017)
50. Maria Carlucci, F., Porzi, L., Caputo, B., Ricci, E., Rota Bulo, S.: Autodial: Automatic domain alignment layers. In: ICCV (2017)
51. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. ECCV (2020)
52. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: ICCV (2015)
53. Pandey, P., Tyagi, A.K., Ambekar, S., Prathosh, A.: Unsupervised domain adaptation for semantic segmentation of nir images through generative latent search. In: ECCV (2020)
54. Peng, X., Li, Y., Saenko, K.: Domain2vec: Domain embedding for unsupervised domain adaptation. arXiv:2007.09257 (2020)
55. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: ECCV (2016)
56. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
57. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
58. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. TPAMI (2017)
59. Shin, I., Woo, S., Pan, F., Kweon, I.S.: Two-phase pseudo label densification for self-training based domain adaptation. In: ECCV (2020)
60. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: NeurIPS (2020)
61. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV (2016)
62. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR (2011)
63. Truong, T.D., Duong, C.N., Le, N., Phung, S.L., Rainwater, C., Luu, K.: Bimal: Bijective maximum likelihood approach to domain adaptation in semantic scene segmentation. In: ICCV (2021)
64. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: CVPR (2018)
65. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: ICCV (2019)
66. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)

67. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv:1412.3474 (2014)
68. Volpi, R., Morerio, P., Savarese, S., Murino, V.: Adversarial feature augmentation for unsupervised domain adaptation. In: CVPR (2018)
69. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
70. Wang, H., Shen, T., Zhang, W., Duan, L.Y., Mei, T.: Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In: ECCV (2020)
71. Wang, Q., Dai, D., Hoyer, L., Van Gool, L., Fink, O.: Domain adaptive semantic segmentation with self-supervised depth estimation. In: ICCV (2021)
72. Wei, G., Lan, C., Zeng, W., Chen, Z.: Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In: CVPR (2021)
73. Woong-Gi, C., Tackgeun, Y., Seonguk, S., Suha, K., Bohyung, H.: Domain-specific batch normalization for unsupervised domain adaptation. In: CVPR (2019)
74. Xie, S., Zheng, Z., Chen, L., Chen, C.: Learning semantic representations for unsupervised domain adaptation. In: ICML (2018)
75. Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., Zhang, W.: Adversarial domain adaptation with domain mixup. In: AAAI (2020)
76. Yang, J., Xu, R., Li, R., Qi, X., Shen, X., Li, G., Lin, L.: An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In: AAAI (2020)
77. Yang, J., An, W., Wang, S., Zhu, X., Yan, C., Huang, J.: Label-driven reconstruction for domain adaptation in semantic segmentation. In: ECCV (2020)
78. Yang, J., Li, C., An, W., Ma, H., Guo, Y., Rong, Y., Zhao, P., Huang, J.: Exploring robustness of unsupervised domain adaptation in semantic segmentation. In: ICCV (2021)
79. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: CVPR (2018)
80. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: CVPR (2020)
81. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
82. Zhang, F., Koltun, V., Torr, P., Ranftl, R., Richter, S.R.: Unsupervised contrastive domain adaptation for semantic segmentation. arXiv:2204.08399 (2022)
83. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: CVPR (2021)
84. Zhang, Q., Zhang, J., Liu, W., Tao, D.: Category anchor-guided unsupervised domain adaptation for semantic segmentation. NeurIPS (2019)
85. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. In: CVPR (2019)
86. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. In: ECCV (2018)
87. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
88. Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C.C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: ECCV (2018)
89. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: ICCV (2019)
90. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV (2018)