Source-Free Domain Adaptation for Semantic Segmentation

Yuang Liu , Wei Zhang*, Jun Wang*
East China Normal University, Shanghai, China
{frankliu624, zhangwei.thu2011, wongjun}@gmail.com

Abstract

Unsupervised Domain Adaptation (UDA) can tackle the challenge that convolutional neural network (CNN)-based approaches for semantic segmentation heavily rely on the pixel-level annotated data, which is labor-intensive. However, existing UDA approaches in this regard inevitably require the full access to source datasets to reduce the gap between the source and target domains during model adaptation, which are impractical in the real scenarios where the source datasets are private, and thus cannot be released along with the well-trained source models. To cope with this issue, we propose a source-free domain adaptation framework for semantic segmentation, namely SFDA, in which only a well-trained source model and an unlabeled target domain dataset are available for adaptation. SFDA not only enables to recover and preserve the source domain knowledge from the source model via knowledge transfer during model adaptation, but also distills valuable information from the target domain for self-supervised learning. The pixel- and patch-level optimization objectives tailored for semantic segmentation are seamlessly integrated in the framework. The extensive experimental results on numerous benchmark datasets highlight the effectiveness of our framework against the existing UDA approaches relying on source data.

1. Introduction

Semantic segmentation has been a critical computer vision task, which aims to segment and parse a scene image into different image regions associated with semantic categories. It is critical for precisely understanding the visual scene and can be applied to numerous potential applications, such as autonomous driving [7], visual grounding [20, 45, 39], and image editing [31]. But the success of current segmentation techniques depends on large-scale densely-labeled datasets that are prohibitively expensive to be collected in reality. For instance, it takes about 90 minutes to manually annotate a Cityscapes image. An intuitive method to address this issue is transferring knowledge from existing models trained on source datasets to the unlabeled target domain. However, it tends to be hindered by the issue of domain shift which is caused by various data distributions in source and target domains.

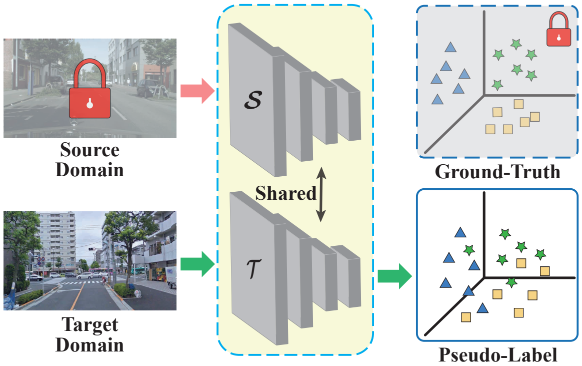


Figure 1. Overview of source-free UDA for segmentation.

Unsupervised domain adaptation (UDA) [13, 54, 19, 6] for semantic segmentation has been proposed to address this issue and generalize the well-trained models on an unlabeled target domain, avoiding expensive data annotation. All the methods suppose that both the well-trained source models and labeled source datasets are available. This is because source data plays a vital role in retaining valuable source knowledge during adaptation training and reducing the cross-domain discrepancy iteratively. However, in some crucial areas like autonomous driving, the source datasets may be private and commercial, making only the source models and unlabeled target datasets available. Due to the lack of supervision of the source domain and the uncertainty of target pseudo-labels, none of these UDA methods can work in such source-free scenarios.

With these insights, we formulate a new but important problem — source-free domain adaptation for semantic segmentation, in which only a well-trained source model and an unlabeled target domain dataset are available for adaptation. Recently, a tiny number of source-free UDA methods [25, 24, 27, 38, 22, 26] have been developed to tackle a similar issue on image classification. However, the image-level computer vision task just associates the label with a whole image, which is fundamentally different from image

[*] Corresponding author.

segmentation that belongs to a pixel-level task with each pixel associated with a semantic label. As shown in Figure 1, the pseudo-labels of one target image contains multiple classes shifting on diverse distributions. As such, it is nontrivial for the above methods to leverage clustering for each class adaptation. Since considering that the source domain knowledge cannot be preserved and utilized without source data, so we attempt to recover and transfer the source domain knowledge by introduced data-free knowledge distillation approaches [29, 3, 30, 11, 48] that are originally for model compression.

In this work, we propose a novel source-free unsupervised domain adaptation framework for segmentation, namely SFDA. Our framework alternatively works in two stages: knowledge transfer and model adaptation. Due to unavailable source data and uncertain target pseudo-labels, recovering and preserving the source knowledge learned by a source model is vital during adaptation training. This is because the uncertain supervision information in target pseudo-labels will tend to deviate the target model from the working domain. As such, in the knowledge transfer stage, we leverage a generator to estimate the source domain (working domain) and synthesize fake samples similar to the real source data in distribution, which can be used to transfer the domain knowledge from a well-trained source model to a target model. The key to semantic segmentation networks lies in capturing contextual feature relationships. With this intuition, a dual attention distillation (DAD) mechanism is introduced to help the generator synthesize samples with meaningful semantic context, which is beneficial to efficient pixel-level domain knowledge transfer. Moreover, the source model could work well on partial target domain and predict correct labels. Therefore we propose an entropy-based intra-domain patch-level self-supervision module (IPSM) to leverage the correctly segmented patches as self-supervision during the model adaptation stage.

Our main contributions can be summarized as follows:

- We propose the novel SFDA framework that combines knowledge transfer and model adaptation without requiring any source data and target labels. To our best knowledge, this is the first attempt to address the problem of source-free UDA for semantic segmentation.
- A novel dual attention distillation mechanism is designed specifically for segmentation to transfer and retain the contextual information, and the intra-domain patch-level self-supervision module is introduced to exploit patch-level knowledge in target domain.
- We demonstrate the effectiveness of our framework on synthetic-to-real and cross-city segmentation scenarios. In particular, it can even achieve competitive results with the state-of-the-art source-driven UDA approaches under the source-free setting.

2. Related Work

UDA for Semantic Segmentation. Existing UDA methods for segmentation can be mainly divided into three categories. To reduce the cross-domain discrepancy, numerous UDA methods [19, 42, 43, 34] focus on distribution consistency by introducing adversarial learning. Inspired by image-to-image translation [21, 54], a category of UDA methods has been proposed to generate target images conditioned on source data [19, 18]. In addition, self-supervision with target pseudo-labels is a relatively simple but efficient approach [6, 55], but it requires source data for supervision. In summary, all the above UDA methods for segmentation assume that the densely-annotated source dataset is available during adaptation, ignoring the data privacy and inaccessibility issues in practice. To the best of our knowledge, we are the first to consider the source-free unsupervised domain adaptation issue for image segmentation.

Knowledge Distillation (KD). Knowledge distillation is originally developed to transfer knowledge from a large teacher network to a compact student network [17]. Since then, a variety of KD methods has been presented for model compression [28, 2, 50, 32], domain adaptation [52, 53], and multi-modal learning [51, 14, 10]. More recently, data-free knowledge distillation [29, 3] has drawn surging attention, due to the inevitable data privacy issue. In [29, 33], activation records are used to reconstruct training samples for training a compact student model. Analogously, Batch Normalization Statistics (BNS) stored in Batch Normalization (BN) layers can be used to reconstruct training samples [48, 15] as well. Most of the data-free KD methods based on generative adversarial networks [3, 49, 30, 11, 47]. They all focus on generating fake samples for transferring knowledge from teacher to student networks without original training data mainly on classification tasks. In this work, we extend the data-free knowledge distillation methods to segmentation and tackle the source-free domain adaptation challenge.

3. Methodology

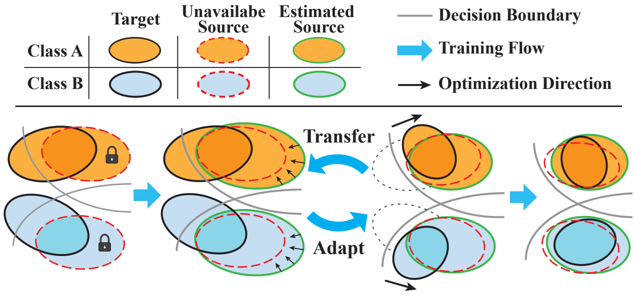


Figure 2. Overview of the training procedure for the proposed framework. Due to the unavailable source domain (marked as red dotted ellipse), we adopt a generator to estimate it by synthesizing fake samples (marked as green ellipse).

3.1. Notations and Motivation

For exiting source-driven UDA methods, an annotated source dataset $D_s = \{(x_s, y_s) | x_s \in \mathbb{R}^{H \times W \times 3}, y_s \in \mathbb{R}^{H \times W}\}$, an unlabeled target dataset $D_t = \{x_t | x_t \in \mathbb{R}^{H \times W \times 3}\}$ and a well-trained source model $\mathcal{S}$ are given. Note that x_s and x_t corresponds to the source and target sample, respectively, and y_s is the label for the corresponding source image. H and W are the height and width of the images. The target model $\mathcal{T}$ generally shares parameters with the source model, but takes target data as input during adaptation. The source-driven UDA methods are commonly formulated by:

$$\mathcal{L}_{DA} = \mathcal{L}_{SEG}(x_s, y_s) + \mathcal{L}_{TAR}(x_t)\,, \tag{1}$$

where $\mathcal{L}_{SEG}$ is the supervised training loss for preserving source domain knowledge, usually cross-entropy or focal loss. And $\mathcal{L}_{TAR}$ is the self-supervision loss for the target domain based on pseudo-labels, such as entropy minimization [43], maximum square loss (MaxSquare) [6], *etc*. In this work, we adopt the maximum square loss as an assistance during adaptation, which is defined as:

$$\mathcal{L}_{TAR}(x_t) = -\frac{1}{HW} \sum_{h,w}^{HW} \sum_{c}^{C} (p_t^{h,w,c})^2\,, \tag{2}$$

where $p_t^{h,w,c}$ is the probability of category c for one target image pixel and C is the number of semantic categories.

In source-free scenarios, the annotated source dataset is unavailable, so the supervised learning process to preserving source knowledge will abort. Fortunately, the source domain knowledge has been permanently retained in the source model. We can consider source-free UDA as a knowledge transfer and adaptation problem, shown in Figure 2. The orange or blue ellipse areas represent the feature space of the source and target domain. Due to the learning bias, the source model can only work well in the source domain, making it necessary to estimate the source domain (marked as green ellipse) and transfer the knowledge to target model during adaptation. Following the above principle analysis, a source-free UDA framework combining knowledge transfer and adaptation is proposed for semantic segmentation.

We denote the estimated source dataset with labels as $\tilde{D_s} = \{(\tilde{x}_s, \tilde{y}_s) | \tilde{x}_s \in \mathbb{R}^{H \times W \times 3}, \tilde{y}_s \in \mathbb{R}^{H \times W}\}$ (corresponding to the green ellipse in Figure 2). Figure 3 shows our SFDA framework, which includes a **Knowledge Transfer** stage and a **Model Adaptation** stage. Note that, to preserve and transfer the source domain knowledge retained in the source model, we need to copy a source model $\tilde{\mathcal{S}}$ and fix its parameters in training. In the transfer stage, generator $\mathcal{G}$ synthesizes fake samples for transferring the source knowledge from the fixed source model $\tilde{\mathcal{S}}$ to $\mathcal{S}$. Moreover, an intra-domain patch-level self-supervision module (IPSM) is introduced to take advantage of information in patch-level pseudo-labels and improve the utilization of target data. We detail the two-stage SFDA in the following.

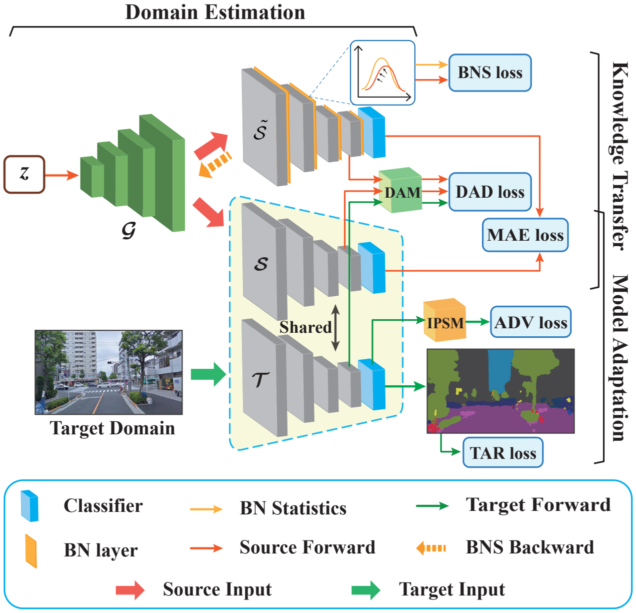


Figure 3. Architecture of the proposed SFDA framework.

3.2. Source-Free Domain Knowledge Transfer

3.2.1 Source Domain Estimation

To estimate the unavailable source domain, a generator $\mathcal{G}$ is designed to generate fake samples $\tilde{x}_s$ with random noises z as input, drawn from a Gaussian distribution.

$$\tilde{x}_s = \mathcal{G}(z),\; z \sim \mathcal{N}(\mathbf{0}, \mathbf{1})\,. \tag{3}$$

Following BNS-guided data-free knowledge distillation [48], the feature distribution of estimated source samples is supposed to satisfy the batch normalization statistics of the source segmentation model. Hence, we apply a BNS constraint on the generator:

$$\mathcal{L}_{BNS} = \sum_{l} \|\mu_l(\tilde{x}_s) - \bar{\mu}_l\|_2^2 + \sum_{l} \|\sigma_l^2(\tilde{x}_s) - \bar{\sigma}_l^2\|_2^2\,, \tag{4}$$

where $\tilde{x}_s$ is the synthetic data from the generator, $\mu_l(\tilde{x}_s)$ and $\sigma_l^2(\tilde{x}_s)$ are the batch-wise mean and variance estimates of feature maps at the l-th layer, and $\bar{\mu}_l$ and $\bar{\sigma}_l^2$ are the corresponding mean and variance parameters of the source domain stored in the l-th BN layer of source model $\tilde{\mathcal{S}}$.

Different from [48], the generative approach for obtaining fake samples in our framework is more efficient and flexible, which avoids the time-consuming noise optimization procedure thanks to the generative adversarial knowledge transfer mechanism. Specifically, for segmentation tasks, we construct a semantic-aware adversarial knowledge transfer mechanism, working based on the discrepancy between the source and target models. To achieve this, we first formulate three different discrepancy measures for three models. The output space discrepancy between the fixed model $\tilde{\mathcal{S}}$ and the shared source model $\mathcal{S}$ is formulated as a mean absolute error (MAE):

$$\mathcal{L}_{MAE} = \mathbb{E}_{\tilde{x}_s} \left(\frac{1}{C} \| \mathcal{S}(\tilde{x}_s) - \tilde{y}_s \|_1 \right) , \tag{5}$$

where $\tilde{y}_s = \tilde{\mathcal{S}}(\tilde{x}_s)$ and $\mathcal{S}(\tilde{x}_s)$ are the prediction outputs from $\tilde{\mathcal{S}}$ and $\mathcal{S}$ for synthetic data $\tilde{x}_s$, respectively.

Moreover, semantic information or contextual relationships performs a significant effect on segmentation. So the contextual relationships captured by the source model are supposed to be preserved and transferred. The discrepancy of the contextual relationships between $\tilde{\mathcal{S}}$ and $\mathcal{S}$ is calculated by a dual attention distillation loss, which is given by:

$$\mathcal{L}_{DAD}^{ss} = \mathbb{E}_{\tilde{x}_s} \left(\frac{1}{M} \| \mathcal{A}(\tilde{\mathcal{F}}^s(\tilde{x}_s)) - \mathcal{A}(\mathcal{F}^s(\tilde{x}_s)) \|_1 \right) , \tag{6}$$

where $\mathcal{A}(\cdot)$ is the dual attention module (DAM) to calculate the dual attention map of the corresponding features. M is the size of the attention map. $\tilde{\mathcal{F}}^s(\tilde{x}_s)$ and $\mathcal{F}^s(\tilde{x}_s)$ are the backbone feature extractors of the segmentation models $\tilde{\mathcal{S}}$ and $\mathcal{S}$ with synthetic data $\tilde{x}_s$ as input.

Analogously, we can define the discrepancy between the source and target models as follows:

$$\begin{aligned} \mathcal{L}_{DAD}^{st} = & \mathbb{E}_{\tilde{x}_s} \left[D_{KL} \left(S(\tilde{\mathcal{F}}^s(\tilde{x}_s)), S(\mathcal{F}^t(x_t)) \right) \right] \\ & + \mathbb{E}_{\tilde{x}_s} \left[D_{KL} \left(R(\tilde{\mathcal{F}}^s(\tilde{x}_s)), R(\mathcal{F}^t(x_t)) \right) \right] , \end{aligned} \tag{7}$$

in which $\mathcal{F}^t(x_t)$ obtains the feature map extracted from the backbone of target model with the target data x_t as input. S and R are the spatial and channel attention maps extracted from the feature maps, which will be defined at Sec 3.2.2. The motivation behind this equation is that the data generated by the generator is not enough to restore the contextual relationships of the source data, due to the lack of necessary prior information. Fortunately, the unlabeled target data has a similar domain-agnostic semantic structure with the real source data to a certain extent. This provides valuable knowledge for the generator to synthesize fake images. So we adopt Kullback-Leibler (KL) divergence to measure the distribution distance of the dual attention maps of fake source and target data, then minimize it in optimization.

3.2.2 Dual Attention Module

In this section, we clarify the dual attention module. The feature map extracted by backbone of segmentation network with x as input is denoted as $F = \mathcal{F}(x), F \in \mathbb{R}^{H_1 \times W_1 \times C_1}$. Note that H_1, W_1, C_1 are the height, width and channel of the feature map respectively only in this sub-section. The dual attention module including spatial attention and channel attention is shown in Figure 4. Different from [12, 46], we feed the feature F into convolutional layers to generate new features, because DAM just aims to capture the spatial and channel-based long-range dependencies for distillation.

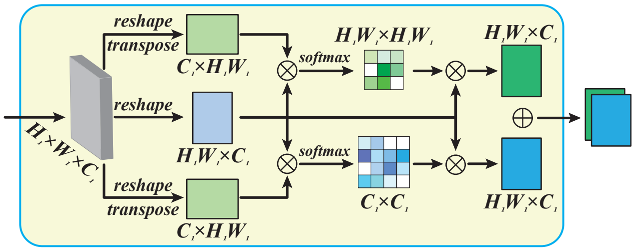


Figure 4. Dual attention module (DAM) for distillation. '$\otimes$' denotes matrix multiplication, and '$\oplus$' is a concatenation operator.

To be specific, we first reshape F so that $F \in \mathbb{R}^{N_1 \times C_1}$, where $N_1 = H_1 \times W_1$ is the number of pixels. $F^\top$ is the transpose of F. Consequently, we calculate the spatial attention map $S \in \mathbb{R}^{N_1 \times N_1}$ by:

$$s_{ji} = \frac{\exp(F_{[i:]} \cdot F_{[:j]}^\top)}{\sum_i^{N_1} \exp(F_{[i:]} \cdot F_{[:j]}^\top)} , \tag{8}$$

where s_{ji} measures the impact of the i-th position on the j-th position.

Analogously, the channel attention map $R \in \mathbb{R}^{C_1 \times C_1}$ can be calculated by:

$$r_{ji} = \frac{\exp(F_{[i:]}^\top \cdot F_{[:j]})}{\sum_i^{C_1} \exp(F_{[i:]}^\top \cdot F_{[:j]})} , \tag{9}$$

where r_{ji} measures the impact of the i-th channel on the j-th channel.

After obtaining the spatial and channel attention maps, the dual attention map of sample x can be calculated by concatenating the two attention maps:

$$\mathcal{A}(x) = \mathtt{concat}(F \cdot S | R \cdot F) . \tag{10}$$

To transform the spatial and channel attention maps to the same shape, they are multiplied by the original feature F, respectively.

3.2.3 Objective Function

In this way, we have introduced all the necessary components for source-free domain knowledge transfer (SFKT). The generator in our framework aims to synthesize valuable fake samples for transferring source knowledge from the source model to the target model. First, it is supposed to make the fake samples comply with the BNS constraints. Second, the generator explores the discrepancy space by maximizing the discrepancy between the source and target models to drive the search for new knowledge. In addition, it's better to take advantage of the prior attention information in the target domain by minimizing $\mathcal{L}^{st}_{DAD}$. Hence, the total objective function of generator is formulated as:

$$\min_{\mathcal{G}} \mathcal{L}_{BNS} - \alpha\mathcal{L}_{MAE} - \beta\mathcal{L}^{ss}_{DAD} + \tau\mathcal{L}^{st}_{DAD}\,, \tag{11}$$

where α, β and τ are hyper-parameters for balancing the MAE loss and the two DAD losses.

The target model learns from two aspects: the target pseudo-labels and two-level knowledge from the source model. We hope that while reducing the uncertainty of the target domain, the target also preserves the source domain information to guide adaptive learning by minimizing the output and attention discrepancy (two-level) with the source model. The objective function of target model in knowledge transfer stage is as follows:

$$\min_{\mathcal{T},\mathcal{S}} \alpha\mathcal{L}_{MAE} + \beta\mathcal{L}^{ss}_{DAD}\,. \tag{12}$$

3.3. Self-supervised Model Adaptation

Since it is hard for the generator to guarantee to continuously restore and transfer the information precisely covering the source domain, we draw inspiration from the self-supervision mechanism and consider taking advantage of the valuable information output by target model for target data. Through analyzing the prediction of the initial target model on the target domain, we found that its prediction on most patches are correct, in which there are useful supervision information for learning on uncertain or error patches.

To take advantage of the pseudo-labels in UDA-based segmentation, Pan *et al*. [34] proposed an unsupervised inter-domain and intra-domain adaptation method, which first separates the target domain into easy and hard splits using an entropy-based ranking function, and then decreases the inter-domain or intra-domain gap via an adversarial mechanism. However, in reality, the gap between the source and the target domain is too large, making it difficult to filter out a sufficient number of easy splits in the target domain for intra-domain supervision. What makes matters worse is that the source domain is unavailable in our setting.

3.3.1 Patch-level Self-supervision Module

To cope with above issue, we present a novel entropy-based intra-domain patch-level self-supervision module to take advantage of the target domain pseudo-labels in the model adaptation stage, shown in Figure 5. Considering in cityscapes segmentation scenarios, there are generally similar patterns or objects in the same areas of different street view images. Hence, we can leverage correct information at the patch level, which not only expands the samples but also alleviates uncertainty in entire pseudo-labels. In order to alleviate the difficulty of separating easy and hard samples caused by too large domain gap [34], we divide each sample into $K \times K$ classes of sub-images or patches with a label k ($k \in \{\mathbb{R}^{K^2}\}$) according to their positions. In prediction, the patches with lower entropy might have higher confidence and accuracy. Hence the patches are split into easy and hard groups by entropy-ranking.

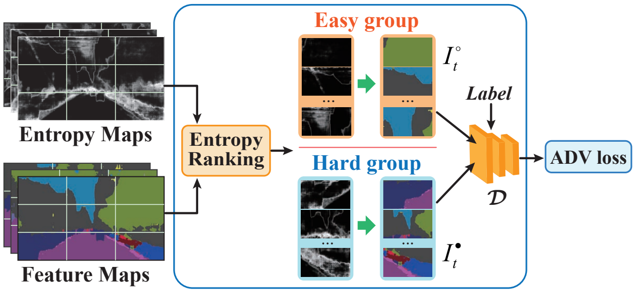


Figure 5. Intra-domain patch-level self-supervision module (IPSM).

We denote the height and width of each patch $x_{t,k}$ in target data $x_t \in \mathbb{R}^{H\times W\times C}$ as $H_2 = H/K$ and $W_2 = W/K$, and the corresponding prediction map output by the target model is $i_{t,k} \in \mathbb{R}^{H_2\times W_2\times C}$, C is both the number of semantic categories and the channel of prediction maps. The probability map $p_{t,k}$ of patch $x_{t,k}$ can be calculated by a `softmax` function.

Then, the mean entropy score of each prediction map $p_{t,k}$ for the target image x_t is defined as:

$$E(x_{t,k}) = -\frac{1}{H_2W_2}\sum_{h,w}^{H_2W_2}\sum_{c}^{C} p_{t,k}^{h,w,c}\log(p_{t,k}^{h,w,c})\,. \tag{13}$$

In a batch containing B (even number) target images $\{x^b_{t,k}|b \in \{1,...,B\}\}$, entropy-ranking is executed on patch entropy maps at the same position or class. The $B/2$ prediction maps in each class with lower entropy are assigned to the easy group $I^{\circ}_{t,k} = \{i^e_{t,k}|e \in \{1,...,B/2\}\}$, while the other $B/2$ are assigned to the hard group $I^{\bullet}_{t,k} = \{i^d_{t,k}|d \in \{1,...,B/2\}\}$. This process is given as follows:

$$I^{\bullet}_{t,k}, I^{\circ}_{t,k} \leftarrow \mathtt{Rank}(\{E(x^b_{t,k})|b \in \mathbb{R}^B\}),\ k \in \mathbb{R}^{K^2}\,. \tag{14}$$

After obtaining the prediction maps of hard and easy patches, we train a discriminator $\mathcal{D}$. $\mathcal{D}$ aims to discriminates easy and hard patches, while $\mathcal{T}$ is trained to fool $\mathcal{D}$ from the side of hard patches to reduce the gap between patches. The adversarial learning loss to optimize $\mathcal{T}$ and $\mathcal{D}$ is given by:

$$\begin{aligned}\mathcal{L}_{ADV}(I^{\bullet}_t, I^{\circ}_t) = -\sum_{k}^{K^2}\sum_{d,e}^{B/2} &\log\left(1-\mathcal{D}(k, i^e_{t,k})\right)\\ &+\log\left(\mathcal{D}(k, i^d_{t,k})\right)\,.\end{aligned} \tag{15}$$

Method	SF	Network	road	sidewalk	building	wall	fence	pole	light	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mIoU
source only	✗	DeepLabV3	74.4	23.3	78.4	27.8	19.6	17.3	18.0	7.4	78.3	24.6	80.5	42.9	6.6	73.2	22.2	29.7	4.2	18.1	1.5	34.09
MinEnt [43]	✗		80.2	31.9	81.4	25.1	20.8	24.6	30.2	17.5	83.2	18.0	76.2	55.2	24.6	75.5	33.2	31.2	4.4	27.4	22.9	40.17
AdaptSegNet [41]	✗		81.6	26.6	79.5	20.7	20.5	23.7	29.9	**22.6**	81.6	26.7	81.2	52.4	20.2	79.1	**36.0**	28.8	**7.5**	24.7	26.2	40.49
CBST [55]	✗		84.8	**41.5**	80.4	19.5	**22.4**	24.7	30.2	20.4	**83.5**	29.6	82.3	54.7	25.3	79.2	34.5	32.3	6.8	29.0	**34.9**	42.94
MaxSquare [6]	✗		**85.8**	33.6	82.4	25.3	25.0	**26.5**	**33.3**	18.7	83.2	**32.9**	79.8	57.8	22.2	**81.0**	32.1	32.6	5.2	**29.8**	32.4	43.12
SFDA (w/o IPSM)	✓		83.5	33.9	81.4	24.8	22.4	23.6	30.1	19.8	81.4	28.7	80.9	56.8	20.4	78.6	35.0	28.9	3.6	26.4	25.5	41.35
SFDA	✓		84.2	39.2	**82.7**	**27.5**	22.1	25.9	31.1	21.9	82.4	30.5	**85.3**	**58.7**	22.1	80.0	33.1	**31.5**	3.6	27.8	30.6	**43.16**
source only	✗	SegNet	48.9	17.2	76.4	6.7	12.5	22.8	12.6	4.8	77.2	15.1	74.2	47.2	7.2	57.7	20.3	10.2	1.0	2.2	1.1	27.13
MinEnt	✗		79.8	31.7	78.8	20.2	18.4	23.9	14.7	4.9	80.6	17.9	78.4	48.9	5.2	77.6	21.7	17.1	**12.7**	**10.5**	2.6	33.97
AdaptSegNet	✗		82.1	29.2	79.4	21.1	17.9	24.1	11.0	**7.1**	**82.0**	26.6	74.9	46.5	6.7	73.5	26.0	18.0	10.5	9.3	3.2	34.16
MaxSquare	✗		**82.9**	33.6	80.2	**22.7**	**20.2**	**26.3**	15.5	6.1	81.8	**27.5**	78.8	48.3	**10.1**	79.8	24.4	20.1	13.2	9.4	5.3	**36.11**
SFDA (w/o IPSM)	✓		80.5	30.3	81.6	24.5	18.0	25.1	13.7	3.2	79.4	25.6	76.3	44.6	7.3	**80.5**	24.7	**21.4**	10.5	4.4	2.5	34.43
SFDA	✓		81.8	**35.4**	**82.3**	21.6	20.2	25.3	**17.8**	4.7	80.7	24.6	**80.4**	**50.5**	9.2	78.4	**26.3**	19.8	11.1	6.7	4.3	35.86

Table 1. Results on GTA5 → Cityscapes. 'SF' represents whether the method is in source-free setting.

3.3.2 Objective Function

Upon this, we extend the objective function in Equation 12 by adding the adversarial loss w.r.t. IPSM and the self-supervision loss. As a result, we define the following objective function to train the target and source models (*i.e.*, $\mathcal{T}$ and $\mathcal{S}$) with shared weights:

$$\min_{\mathcal{T},\mathcal{S}} \max_{\mathcal{D}} \mathcal{L}_{TAR} + \alpha \mathcal{L}_{MAE} + \beta \mathcal{L}_{DAD}^{ss} + \gamma \mathcal{L}_{ADV} \ , \quad (16)$$

where γ is the hyper-parameter to control the adversarial loss. The detailed training algorithm is presented in the supplementary material.

4. Experiments

4.1. Experimental Settings

4.1.1 Datasets and Metrics

Datasets We evaluate our SFDA framework on semantic segmentation under two different settings: synthetic-to-real and cross-city. For the former setting, we follow previous work [55, 41] by considering Cityscapes [8] as the target domain, and GTA5 [36] or SYNTHIA [37] as the source domain. For the latter setting, Cityscapes dataset is used as the source domain and NTHU [44] dataset is as the target domain.

Cityscapes [8] provides 3,975 images with fine-grained segmentation annotations. The synthetic dataset GTA5 [36] contains 24,966 annotated images with a resolution of 1,914×1,052 taken from the GTA5 game. SYNTHIA [37] is used as another synthetic dataset, which contains 9,400 fully annotated 1,280×760 RGB images. The NTHU dataset [44] contains four different cities: Rio, Rome, Tokyo, and Taipei.

Metrics The semantic segmentation performance is evaluated on every category using Intersection-over-Union (IoU) ratio and Pixel-Accuracy (PA). For the whole test set, we calculate Mean Intersection-over-Union (mIoU) and Mean Pixel-Accuracy (mPA).

4.1.2 Implementation Details

Two kinds of segmentation networks are adopted in our experiments. One is DeepLabV3 [5] with the ResNet-50 [16] pre-trained on ImageNet [9], and the other is SegNet [1] with the pre-trained VGG-16 [40] backbone. Considering SegNet in an encoder-decoder architecture, the DAM is connected behind the encoder. When calculating the dual attention maps of target images, an adaptive pooling is applied before DAM. For the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$, we use an architecture similar to [35] but extend $\mathcal{D}$ to a conditional version. The input channel of $\mathcal{D}$ is set to be consistent with the output channel of prediction maps. The latent space dimension for $\mathcal{G}$ and label embedding dimension for $\mathcal{D}$ both are 256. The architectures of the generator and the discriminator are detailed in the supplementary material.

We implement the proposed framework using the PyTorch toolbox on two GTX 2080Ti GPUs. To train the segmentation networks, we use the Stochastic Gradient Descent (SGD) optimizer with Nesterov acceleration where the momentum is 0.9 and the weight decay is 10^{-4}. The initial learning rate is set to 2.5×10^{-4} and is decreased using the polynomial decay with a power of 0.9 as mentioned in [4]. For training the generator and discriminator, Adam optimizer [23] with an initial learning rate of 0.1 is adopted. Due to the difficulty in generating high-resolution images, we resize the images to 512×256 for all datasets. Thanks to full-convolutional segmentation networks, we can set the resolution of synthetic samples to 256×128, which is lower than target data but enough for transferring knowledge. To get a high-quality source model for adaptation, we pre-train the source models for 30 epochs on Cityscapes while for 20 epochs on GTA5 or SYNTHIA. In source-free adaptation, the target model, the generator, and the discriminator are jointly trained on a target domain for 120 epochs with a batch size of 8.

As for hyper-parameters, α and β are set to 1.0 and 0.5 by default, respectively. Notably we set $\tau = \beta$ to balance two DAD losses. We set γ to 0.01 in all experiments if not particularly indicated. The number of patches, *i.e.*, K in IPSM is reasonable to choose from $\{3, 4, 5\}$.

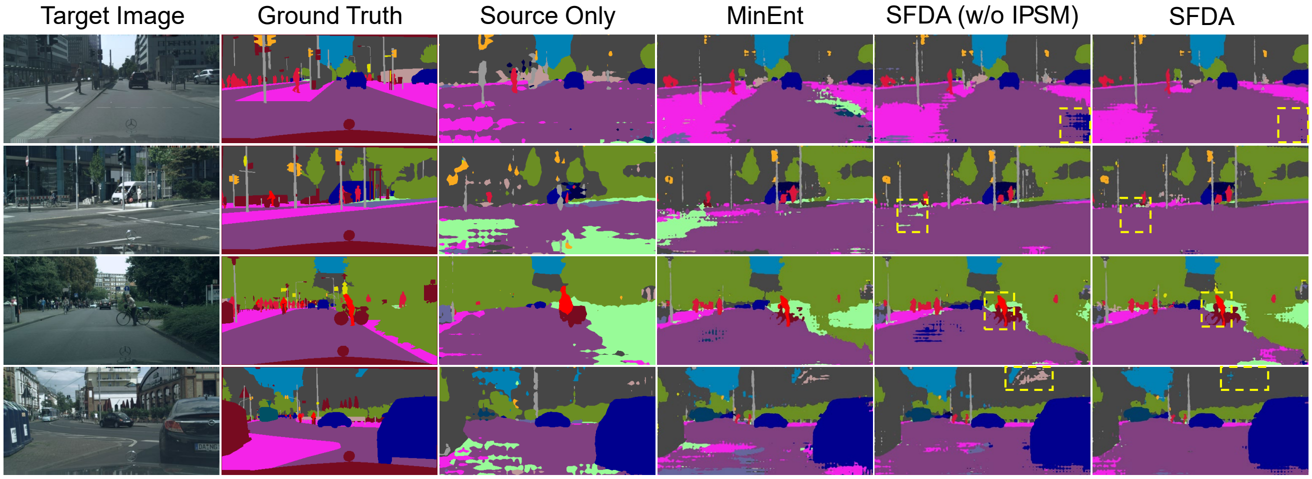


Figure 6. Qualitative results on GTA5 → Cityscapes.

Method	SF	road	sidewalk	building	wall*	fence*	pole*	light	sign	veg.	sky	person	rider	car	bus	motor	bike	mIoU	mIoU*
source only	✗	55.5	21.0	75.8	2.2	0.1	19.9	0.2	6.9	74.5	81.5	37.1	8.7	45.1	18.2	2.5	19.8	29.31	34.36
MinEnt [43]	✗	78.2	39.6	**81.9**	4.3	0.2	**26.2**	2.2	4.1	81.1	87.7	37.7	7.2	75.8	24.9	4.6	25.1	36.30	42.31
AdaptSegNet [41]	✗	79.7	38.6	79.3	5.6	**0.8**	25.4	3.6	5.5	80.0	85.4	**40.8**	11.7	79.8	21.4	5.2	30.5	37.08	43.19
CBST [55]	✗	81.4	44.2	80.4	7.9	0.7	25.6	5.2	12.4	81.4	89.5	39.7	10.6	82.1	21.9	6.3	**32.9**	38.88	45.23
MaxSquare [6]	✗	81.0	39.8	82.6	**8.7**	0.5	23.2	**6.6**	**12.4**	85.3	**90.1**	39.9	8.4	**84.7**	19.4	**10.2**	33.4	39.12	45.65
SFDA(w/o IPSM)	✓	81.5	43.5	80.6	1.4	0.7	19.9	4.2	7.1	83.1	87.6	36.8	9.5	81.3	22.7	8.6	31.7	37.50	44.47
SFDA	✓	**81.9**	**44.9**	81.7	4.0	0.5	26.2	3.3	10.7	**86.3**	89.4	37.9	**13.4**	80.6	**25.6**	9.6	31.3	**39.20**	**45.89**

Table 2. Results of domain adaptation task SYNTHIA → Cityscapes. 'mIoU' and 'mIoU*' are calculated over 16 and 13 classes, respectively.

4.2. Comparison

Synthetic-to-Real Adaptation: (1) GTA5 → Cityscapes. Figure 6 shows the qualitative results on GTA5 → Cityscapes. In order to show the versatility of SFKT and the contribution of IPSM, we remove the IPSM part in our architecture, namely 'SFDA (w/o IPSM)'. It is obvious that even without source data, our method outperforms traditional MinEnt method. What's more, with the enhancement of IPSM, our full method can make up for errors in some areas through self-supervision, shown in the yellow dashed box. We present adaptation results in Table 1 with comparisons to the state-of-the-art source-driven domain adaptation methods.

(2) SYNTHIA → Cityscapes. Following the evaluation setting in [43, 55], we present the results of IoU and mIoU w.r.t. 16-class and 13-class segmentation in Table 2, respectively. Our architecture is used with DeepLabV3, and even outperforms the source-driven UDA methods with the assistance of IPSM. Besides, our method achieves competitive performance for the small object segmentation, such as traffic light, traffic sign, and motorbike.

Cross-City Adaptation: To show the effectiveness of our methods for smaller domain shift, we conduct the experiment on Cityscapes → NTHU with DeepLabV3 architecture. Table 3 shows the comparisons of our method with other source-driven UDA methods. Compared to the best UDA method MaxSquare, our method with IPSM achieves competitive performance on four city datasets. In addition, we distill source domain knowledge via SFKT from well-trained source model into a new model and evaluate it on target domain without adaptation, shown as 'transfer only' in the table. The results demonstrate that the knowledge we obtained via SFKT is still valuable on the target, although the effect is not as good as 'source only'.

Method	SF	Rome	Rio	Tokyo	Taipei
source only	✗	46.44	45.06	44.05	44.07
MinEnt [43]	✗	47.29	46.82	45.49	45.12
AdaptSegNet [41]	✗	47.99	47.81	46.22	45.13
MaxSquare [6]	✗	**48.48**	48.74	**47.10**	47.16
transfer only	✓	45.87	44.03	43.96	43.55
SFDA (w/o IPSM)	✓	47.38	47.75	45.18	45.38
SFDA	✓	48.33	**49.03**	46.36	**47.20**

Table 3. Results on Cross-City adaptation.

4.3. Ablation Study

To show the detailed contributions of the components in SFKT, we conduct ablation experiments on three datasets, shown in Table 4. The results demonstrate that the DAD losses in source-free domain knowledge transfer is more

Dataset	source model	BNS	DAD	BNS+DAD
GTA5 [36]	61.8	49.8	55.4	58.3
Cityscapes [8]	73.6	60.6	65.5	70.8
SYNTHIA [37]	62.3	51.4	54.7	59.0

Table 4. Results for key components in SFKT.

effective than the commonly used BNS loss, and the fusion of them could further improve the performance.

The visualization of semantic maps and fake samples synthesized in the knowledge transfer stage are shown in Figure 7. The left two columns are the fake samples synthesized by generator and corresponding semantic maps predicted by DeepLabV3 pre-trained on Cityscapes. The right two columns are several semantic maps predicted without DAD or BNS loss. On one hand, the output semantic maps are similar to the real-world street view structure without DAD, but it is hard to pay attention to some small objects or refined segmentation. On the other hand, the generator captures the discrepancy between two models, but cannot preserve the original semantic distribution of source domain without BNS loss, which is vital for segmentation tasks. Although the fake samples cannot be recognized by humans, they have similar representations and outputs in convolutional neural networks with the source domain data. Hence, the fake samples become the key to transfer source domain knowledge.

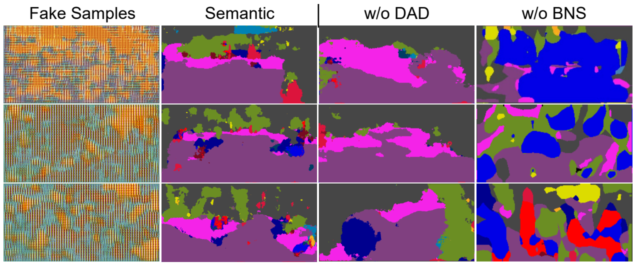


Figure 7. Visualization of synthetic semantic maps.

4.4. Hyper-parameter Analysis

Firstly, we discuss the influence of α and β ($\tau = \beta$), the weights for the MAE loss and the DAD losses, respectively, for DeepLabV3 on GTA5 → Cityscapes. Given $\beta = 0.5$, we adjust α from 0.1 to 2.0, and show the results in Table 5. Since the MAE loss $\mathcal{L}_{MAE}$ of the source prediction output is similar to the target segmentation loss $\mathcal{L}_{TAR}$ when supervised by target pseudo-labels, α should be close to 1.0. Otherwise, there will be disagreements with $\mathcal{L}_{MAE}$, resulting in bias during adaptation.

α	0.1	0.5	1.0	2.0
mIoU	41.33	42.70	43.16	42.47

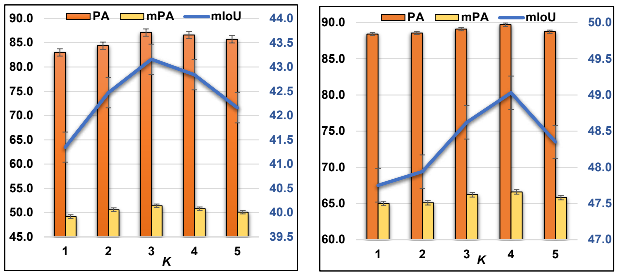
Table 5. Influence of α given $\beta = 0.5$.

Analogously, given $\alpha = 1.0$, we adjust β from 0.01 to 1.0, and the results are shown in Table 6. Different from α, β controls the weights of the DAD losses in intermediate layers, so they are supposed to be smaller than α. If too many weights are allocated to the DAD losses, they will limit the learning capacity of the intermediate layers.

β	0.01	0.1	0.5	1.0
mIoU	41.54	43.09	43.16	42.47

Table 6. Influence of β given $\alpha = 1.0$.

We show the sensitivity analysis of parameters $K \in \{1, 2, \cdots, 5\}$ in Figure 8, from which we observe that too large or too small K is not suitable for IPSM, and 3 to 5 is reasonable. Note that when $K = 1$, it means IPSM is not adopted in training.



(a) GTA → Cityscapes (b) Cityscapes → Rio

Figure 8. Influence of number of patches (*i.e.*, K).

5. Conclusion

In this paper, we have presented a novel source-free domain adaptation framework (SFDA) for semantic segmentation. It aims to preserve the source domain knowledge from a fixed source model via knowledge transfer. Specifically, a dual attention distillation method is designed to capture and transfer pixel-level semantic information for segmentation tasks. Moreover, during model adaptation, an intra-domain patch-level self-supervision mechanism is introduced to take advantage of valuable knowledge at patch-level pseudo-labels in a target domain. We conduct extensive experiments and ablation studies to validate the effectiveness of the proposed framework on different segmentation tasks, showing it performs favorably against existing source-driven UDA methods. However, our approach does not support high-resolution image segmentation tasks due to the limitation of generative fake sample synthesis, which will be tackled in future work.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China under Grant (No. 62072182).

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI*, 39(12):2481–2495, 2017.

[2] Shu Changyong, Li Peng, Xie Yuan, Qu Yanyun, Dai Longquan, and Ma Lizhuang. Knowledge squeezed adversarial network compression. In *AAAI*, 2020.

[3] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *ICCV*, pages 3514–3522, 2019.

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

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *ICCV*, pages 2090–2099, 2019.

[7] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, pages 7892–7901, 2018.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[10] Qi Dou, Quande Liu, Pheng Ann Heng, and Ben Glocker. Unpaired multi-modal segmentation via knowledge distillation. *arXiv preprint arXiv:2001.03111*, 2020.

[11] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. In *CVPR*, 2020.

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

[14] Nuno C Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. In *ECCV*, pages 103–118, 2018.

[15] Matan Haroush, Itay Hubara, Elad Hoffer, and Daniel Soudry. The knowledge within: Methods for data-free model compression. In *CVPR*, pages 8494–8502, June 2020.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1989–1998, 2018.

[19] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *CVPR*, pages 1335–1344, 2018.

[20] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, pages 108–124. Springer, 2016.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[22] Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. Domain adaptation without source data. *arXiv preprint arXiv:2007.01524*, 2020.

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

[24] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *CVPR*, pages 4544–4553, 2020.

[25] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, pages 9641–9650, 2020.

[26] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *CVPR*, pages 2975–2984, 2019.

[27] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

[28] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *CVPR*, pages 7096–7104, 2019.

[29] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[30] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *NeurIPS*, pages 9551–9561, 2019.

[31] J-M Morel, Ana Belén Petro, and Catalina Sbert. Fourier implementation of poisson image editing. *Pattern Recognition Letters*, 33(3):342–348, 2012.

[32] Subhabrata Mukherjee and Ahmed Awadallah. Tinymbert: Multi-stage distillation framework for massive multi-lingual ner. In *ACL*, 2020.

[33] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, R Venkatesh Babu, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *ICML*, 2020.

[34] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, pages 3764–3773, 2020.

[35] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[36] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, pages 102–118. Springer, 2016.

[37] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, pages 3234–3243, 2016.

[38] Roshni Sahoo, Divya Shanmugam, and John Guttag. Unsupervised domain adaptation in the absence of source data. *arXiv preprint arXiv:2007.10233*, 2020.

[39] Gunnar A Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, João Carreira, Phil Blunsom, and Andrew Zisserman. Visual grounding in video for unsupervised word translation. In *CVPR*, pages 10850–10859, 2020.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[41] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pages 7472–7481, 2018.

[42] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *ICCV*, pages 1456–1465, 2019.

[43] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, pages 2517–2526, 2019.

[44] Ching-Hua Weng, Ying-Hsiu Lai, and Shang-Hong Lai. Driver drowsiness detection via a hierarchical temporal deep belief network. In *ACCV*, pages 117–133. Springer, 2016.

[45] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *CVPR*, pages 5945–5954, 2017.

[46] Jinyu Yang, Weizhi An, Chaochao Yan, Peilin Zhao, and Junzhou Huang. Context-aware domain adaptation in semantic segmentation. *arXiv preprint arXiv:2003.04010*, 2020.

[47] Jingwen Ye, Yixin Ji, Xinchao Wang, Xin Gao, and Mingli Song. Data-free knowledge amalgamation via group-stack dual-gan. In *CVPR*, pages 12516–12525, June 2020.

[48] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *CVPR*, pages 8715–8724, 2020.

[49] Jaemin Yoo, Minyong Cho, Taebum Kim, and U Kang. Knowledge extraction with no observable data. In *NeurIPS*, pages 2701–2710, 2019.

[50] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *CVPR*, pages 3517–3526, 2019.

[51] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *CVPR*, pages 7356–7365, 2018.

[52] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. *arXiv preprint arXiv:1911.11554*, 2019.

[53] Brady Zhou, Nimit Kalra, and Philipp Krähenbühl. Domain adaptation through task distillation. *arXiv preprint arXiv:2008.11911*, 2020.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *ICCV*, 2017.

[55] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018.