Content-Consistent Matching for Domain Adaptive Semantic Segmentation

Guangrui Li[1⋆], Guoliang Kang[2⋆], Wu Liu[3], Yunchao Wei[1], and Yi Yang[1]

[1] ReLER, Centre for Artificial Intelligence, University of Technology Sydney
guangrui.li-1@student.uts.edu.au
{yunchao.wei, yi.yang}@uts.edu.au
[2] Carnegie Mellon University gkang@andrew.cmu.edu
[3] JD AI Research liuwu1@jd.com
[GitHub]

Abstract. This paper considers the adaptation of semantic segmentation from the synthetic source domain to the real target domain. Different from most previous explorations that often aim at developing adversarial-based domain alignment solutions, we tackle this challenging task from a new perspective, *i.e.*, content-consistent matching (CCM). The target of CCM is to acquire those synthetic images that share similar distribution with the real ones in the target domain, so that the domain gap can be naturally alleviated by employing the content-consistent synthetic images for training. To be specific, we facilitate the CCM from two aspects, *i.e.*, semantic layout matching and pixel-wise similarity matching. First, we use all the synthetic images from the source domain to train an initial segmentation model, which is then employed to produce coarse pixel-level labels for the unlabeled images in the target domain. With the coarse/accurate label maps for real/synthetic images, we construct their semantic layout matrixes from both horizontal and vertical directions and perform the matrixes matching to find out the synthetic images with similar semantic layout to real images. Second, we choose those predicted labels with high confidence to generate feature embeddings for all classes in the target domain, and further perform the pixel-wise matching on the mined layout-consistent synthetic images to harvest the appearance-consistent pixels. With the proposed CCM, only those content-consistent synthetic images are taken into account for learning the segmentation model, which can effectively alleviate the domain bias caused by those content-irrelevant synthetic images. Extensive experiments are conducted on two popular domain adaptation tasks, *i.e.*, GTA5→Cityscapes and SYNTHIA→Cityscapes. Our CCM yields consistent improvements over the baselines and performs favorably against previous state-of-the-arts.

Keywords: semantic segmentation, domain adaptation

⋆ indicates equal contribution.

1 Introduction

Semantic segmentation [4,31,54,29,5,20,9] plays an important role in many real-world applications. However, in practice, the off-the-shelf segmentation model trained on one scenario (source) usually cannot generalize well to the new one (target). For example, for the self-driving task, we may collect data and train a segmentation model in one city, but such a model may fail to give accurate pixel-level predictions for the scenes of another unfamiliar city. As achieving massive in-domain pixel-level annotations are expensive and sometimes impossible, practitioners usually resort to domain adaptive training to achieve satisfactory results on the target.

Generally, a domain adaptive segmentation model is established on the labeled source images and the unlabeled target images. And the training tries to utilize the knowledge learned from the source and mitigate the domain shift. Previous methods usually achieved the adapted model through the adversarial training [41,44,34,24,13,18,2,27] or by self-training [58,59,28,50]. All those methods employed the entire source domain data throughout the training process, which neglects the fact that not all the source images could contribute to the improvement of adaptation performance, especially at certain training stages. We empirically find that there usually exist "negative" source images which may even harm the adaptation. As shown in Fig. 1, compared to the positive source images, the negative ones appear quite dissimilar to the target images. Moreover, from Fig. 1, we observe that for the visually similar pair of source (*i.e* positive source image) and target images, the pixel-wise similarities vary a lot spatially, which implies that the pixels of a source image also should not be treated equally.

In this paper, we propose Content-Consistent Matching (CCM) to match and select the effective source information actively to facilitate the adaptation process. To be specific, we perform *Semantic Layout Matching* to select the positive source samples and *Pixel-wise Similarity Matching* to emphasize effective pixels. For the Semantic Layout Matching, we propose a novel image representation that encodes the semantic layout information. Based on such semantic layout representation, we perform clustering on the target to discover the underlying patterns in the target domain. Then the source sample is selected as the positive one if it is close enough to these patterns. Moreover, we further select the positive source pixels to mitigate the negative transfer through proposed pixel-wise similarity matching. Similar to the matching strategy in the image level, pixel-wise similarity matching selects the source pixels that share similar feature distributions with the target samples.

As the target feature evolves during training, the same source sample may contribute differently to the adaptation process, *e.g.* a source sample could be negative before a certain stage but positive afterward. Thus we choose to iteratively update the matching results during training to enable more effective adaptation. Specifically, we perform the CCM along with a self-training paradigm, *i.e.* we alternatively update the representations through self-training and the

Fig. 1: Examples of positive and negative source samples (Best viewed in color). Generally, positive source images (c) share similar layout with the target (b) while the negative source ones (a) do not. Intuitively, samples like (c) should be selected and samples like (a) should be excluded to help the adaptation. Moreover, the heatmap (d) indicates the pixel-wise similarities between target and positive source embeddings, in which red indicates higher similarity. It can be seen that the similarities vary a lot, even for the semantic-consistent pixels of the source image, which implies the pixels of positive source images should not be equally treated. Detailed information could be found in Sec. 3.2.

source matching results through CCM. These two parts depend on each other and cooperate to mitigate the domain shift.

In a nutshell, our contributions are three-fold:

- We deal with the domain adaptive segmentation task from a new perspective, *i.e.* actively selecting positive source information for training to avoid negative transfer, which has not been investigated by previous methods.
- We propose Content-Consistent Matching (CCM), which consists of Semantic Layout Matching and Pixel-wise Similarity Matching, to select the positive source samples and their positive pixels to facilitate the adaptation process.
- Experiments on two representative benchmarks (*i.e.* GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes) demonstrate that our method performs favorably against previous methods. Ablation studies also verify the effectiveness of the key components of our framework.

2 Related Works

Unsupervised Domain Adaptation(UDA) Unsupervised domain adaptation aims to minimize the distribution discrepancy between the source domain and the target domain. To achieve this goal, some earlier [32,31,39,43] methods proposed to learn domain invariant features via directly minimizing the discrepancy of feature distribution. More recent approaches [33,16,19,57,23,14]

employed adversarial training in image level. [2,15,30,42,48,45] leveraged adversarial training to learn domain-invariant representations in feature level. There are some works [38,51,2,49,22] using self-training to mitigate the domain gap via assigning labels to the most confident samples in the target domain.

Self-Training Self-training, which assigns and updates pseudo labels in an alternative style, has attracted wide attention for its simplicity and effectiveness. Self-training has been exploited in various tasks such as semi-supervised learning [25,21], domain adaptation [38,58], and noisy label learning [40,35]. Most existing UDA methods established on self-training mainly focused on how to utilizing the pseudo labels in the training while neglecting the selection of source samples.

UDA for semantic segmentation Unsupervised domain adaptation for semantic segmentation is the task that applies domain adaptation at pixel-level. Many approaches [53,52,2,10,16] have been proposed. There are mainly two ways to tackle this problem, *i.e.* via adversarial training or self-training. The works that exploited adversarial training can be categorized into the feature-level adaptation and the image level adaptation. Some works [41,44,34,47,24,13] adopted adversarial training at feature level to learn domain-invariant features to reduce the discrepancy across domains. [18,8,27] applied adversarial training at the image level to make features invariant to illumination, color and other style factors. Some recent approaches adopted self-training to perform adaptation. [58] proposed to assign pseudo labels in a curriculum way and [50,59,28,55] combined self-training with other constraints to improve the quality of pseudo labels.

3 Content-Consistent Matching

We aim to train a segmentation network with parameters θ to give accurate pixel-level predictions $P(c|x, y; I^T, \theta)$ on the target T, where $c \in \{0, 1, \cdots, C-1\}$ denotes the underlying categories and $x \in \{0, 1, \cdots, W-1\}, y \in \{0, 1, \cdots, H-1\}$ are the horizontal and vertical coordinates of a pixel in a target image I^T, respectively. The segmentation network is trained with the combination of labeled source images D^S and unlabeled target images D^T. During training, we propose to use content-consistent matching (CCM) to select positive source samples and their effective pixels. Our CCM is performed upon the self-training paradigm, *i.e.* with selected positive source samples and their effective pixels, the network is trained with ground-truth source labels and pseudo target labels to update the feature representation, and based on the updated feature representation, the set of positive source samples and pixels are reconstructed.

3.1 Semantic Layout Matching

Semantic layout means how the categories are distributed spatially in an image (*i.e.* $P(x, y|c)$). It could be an important prior during the training of segmentation models. However, the semantic layout patterns may vary a lot across

domains, leading to the domain shift and degenerating the generalization. For example, it is natural that part of the source domain images is captured from a distinct perspective compared to the target. Thus the semantic layout of these source images will be quite different from the target. In this section, we propose using semantic layout matching to select the positive source samples to mitigate such domain shift.

Semantic Layout Matrix (SLM) Directly using $P(x, y|c)$ to model the semantic layout would be inefficient due to its high dimension, and ineffective because it is not robust to the inaccurate target predictions.

Following the naive Bayes assumption, we propose to decouple $P(x, y|c)$ into the horizontal one $P(x|c)$ and vertical $P(y|c)$ one, *i.e.*,

$$P(x, y|c) \propto P(x|c)P(y|c). \tag{1}$$

Specifically, take the vertical distribution $P(y|c)$ for an example, $P(y|c)$ can be represented as

$$P(y|c) = \frac{P(c|y)P(y)}{P(c)} = \frac{\sum_x P(c|x, y)P(x)P(y)}{\sum_x \sum_y P(c|x, y)P(x)P(y)}, \tag{2}$$

Assuming $P(x)$ and $P(y)$ are the uniform distributions, *i.e.* $P(x = i) = \frac{1}{W}, i \in \{0, 1, \cdots, W-1\}$ and $P(y = j) = \frac{1}{H}, j \in \{0, 1, \cdots, H-1\}$, then

$$P(y|c) = \frac{\sum_x P(c|x, y)}{\sum_x \sum_y P(c|x, y)}. \tag{3}$$

For the source image, suppose its ground-truth label is c', $P(c|x, y; I^S) = \begin{cases} 1 \text{ if } c = c' \\ 0 \text{ otherwise} \end{cases}$. For the target images, as we don't know the ground-truth pixel-wise labels, we adopt the probability predictions $P(c|x, y; I^T, \theta)$ of current segmentation model with parameters θ to compute Eq. (3).

Following the general practice, the images (source and target) are customized as the same size during training, and the vertical semantic layout matrix $M_v \in \mathbb{R}^{C \times H}$ can be expressed as (we omit the domain subscript for simplification)

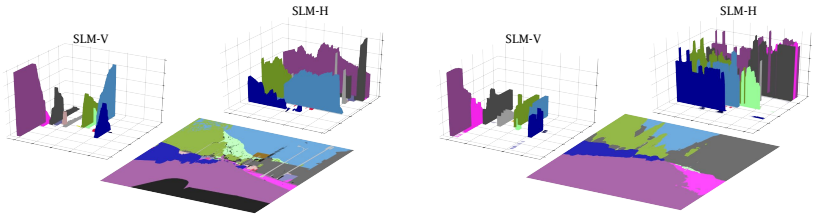
$$M_v(\hat{c}, j) = \frac{\sum_x P(\hat{c}|x, y = j)}{\sum_x \sum_y P(\hat{c}|x, y)}. \tag{4}$$

Similarly, the horizontal semantic layout matrix $M_h \in \mathbb{R}^{C \times W}$ is

$$M_h(\hat{c}, i) = \frac{\sum_y P(\hat{c}|x = i, y)}{\sum_x \sum_y P(\hat{c}|x, y)}. \tag{5}$$

Finally, the semantic layout matrix $M \in \mathbb{R}^{C \times (H+W)}$ can be represented as

$$M = \left[M_h, M_v \right]. \tag{6}$$



(a) Target Prediction and its SLM (b) Positive Source Label and its SLM

Fig. 2: Illustration of SLM (Best viewed in color). Taking the class sky (annotated with sky blue) as an example, we explain how to use SLM to represent the spatial distribution. From SLM-V, we could know its vertical distribution: most of the pixels belonging to the sky are at the top. Through SLM-H, we could also found that the sky mostly lies in the middle and right of the images. (a) and (b) are pairs that share most similar spatial distributions.

Note that because the assumption in Eq. (1) may not be exactly satisfied in practice, we choose to concatenate the horizontal and vertical semantic layout matrix together rather than multiply them, which makes the training less dependent on the conditional independence assumption.

Matching and selection Based on the proposed SLM, we can encode the semantic layout of each target image. We then adopt k-means clustering to discover the underlying K patterns of target SLMs. The source image, which is close enough to these patterns, can be viewed as a positive sample. Note that because we perform single-direction selection, clustering on the source images is not needed.

Specifically, we denote the centers of the K target clusters as $\hat{M}^{T,k}, k \in \{0, 1, \cdots, K-1\}$ and compute the similarity between the source sample and each of these cluster centers through

$$Sim(M^S, \hat{M}^{T,k}) = -\sum_c D_{KL}(\hat{M}_h^{T,k}(c,:)||M_h^S(c,:)) + D_{KL}(\hat{M}_v^{T,k}(c,:)||M_v^S(c,:)), \tag{7}$$

where the $D_{KL}(\cdot||\cdot)$ denotes the KL divergence. And the similarity score of a source image is

$$Score(M^S) = \frac{1}{K}\sum_k Sim(M^S, \hat{M}^{T,k}). \tag{8}$$

Based on the ranking of above similarity scores among all the source samples, only the top ranking source samples are selected for training. In our experiment, we set the selection proportion γ_{img} as a hyper-parameter to control the number of selected source images. We will discuss this further in our experiment part. Our proposed SLM is illustrated in Fig. 2.

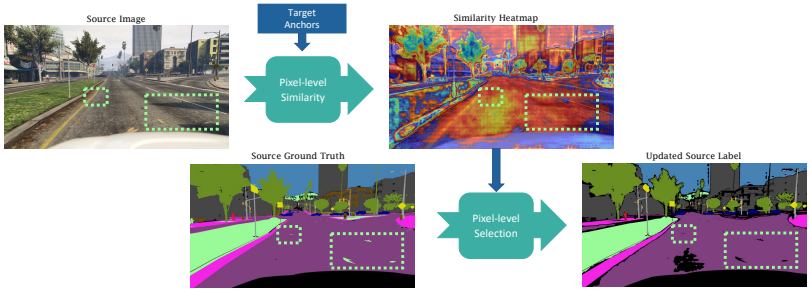


Fig. 3: Illustration of pixel-wise similarity matching. As marked by the green box, the leaves on the road are hardly spotted even by a human but annotated in the ground truth. Pixel-wise similarity matching excludes these pixels which may hinder the adaptation. The black area in the figure denotes the ignored pixels. Best viewed in color.

3.2 Pixel-wise Similarity Matching

For a source image, it is possible that partial regions or pixels are similar to the target, while others not. That means the pixels in a source image should not be equally treated during the adaptation. Thus besides selecting positive source samples, we propose to select the positive source pixels that share similar characters with the target to mitigate the domain shift further. We name such pixel-level selection as pixel-wise similarity matching, as illustrated in Fig. 3.

For a target image, based on the network's outputs $P(c|x,y;I^T,\theta)$, we could assign a pseudo label to each pixel, *i.e.*

$$L^T(x,y) = \arg\max_c P(c|x,y;I^T,\theta). \tag{9}$$

Then the pixels are classified into C groups. The pixel with low confidence prediction $P(L^T(x,y)|x,y;I^T,\theta)$ is filtered (see Section 3.3 for more details). And the average class distribution is calculated among each group

$$Q^T(c) = \frac{1}{|D^T|}\sum_i \frac{1}{|G^T_{i,c}|}\sum_{(\hat{x},\hat{y})\in G^T_{i,c}} P_i(c|\hat{x},\hat{y};I^T_i,\theta), c\in\{0,1,\cdots,C-1\}, \tag{10}$$

where $G^T_{i,c} = \{(\hat{x},\hat{y})|L^T_i(\hat{x},\hat{y}) = c\}$, $Q^T(c)\in\mathbb{R}^C$, and the subscript i denotes the i-th target sample here, $|D^T|$ is the number of samples within target domain. By this way, it is expected that $Q^T(c)$ could describe the relationships between class c and all the other classes, based on the current predictions of the network.

From the network, we can also get the predictions for the source image, *i.e.* $P(c|x,y;I^S,\theta)$, thus it is natural to select such source pixels $\{(\tilde{x},\tilde{y})\}$ where $P(c|\tilde{x},\tilde{y};I^S,\theta)$ matches well with Q^T. We adopt KL divergence to measure the distance between each pair of $P(c|x,y;I^S,\theta)$ and $Q^T(c)$. And the matching score

for a source pixel at (x, y) with ground-truth label c is computed as

$$Score(x, y) = -D_{KL}(Q^T(c)||P(c|x, y; I^S, \theta)). \tag{11}$$

We rank the source pixels within the same ground-truth class according to their similarity score and select the top ranking pixels for each class. In our experiment, we select the same proportion of pixels γ_{pix} for each class.

3.3 Active Matching with Self-training

As the target feature evolves, the effect of the same source sample on the adaptation process may be different. In this paper, we choose to update the source matching results actively throughout the adaptation process. Notice that purely training with the source data may lead to the model biased towards the source distribution, we choose to employ our matching strategy along with the self-training paradigm.

To obtain a good initialization of target predictions, we start the self-training from the segmentation network trained on all the labeled source images D^S. Then we alternatively update the network parameters θ and assign pseudo labels $L^T(x, y)$ on the target D^T according to Eq. (9).

Through pseudo labeling, the target pixels are grouped into C classes. For each class of pixels, we rank them according to the prediction confidences (*i.e.* $P(L^T(x, y)|x, y; I^T, \theta)$). Only the top ranking target pixels are selected for training, and the ratio of selection is set to r, which is shared among all the classes. To enable each target sample to have enough selected pixels, we also perform pixel ranking within each image. Then the top r pixels of a target image are also selected.The selected pixels $\{(\hat{x}, \hat{y})\}$ are assumed to have reliable pseudo labels.

The positive source samples $\hat{D}^S$ selected through our matching strategy, together with the pseudo-labeled target samples D^T, are adopted to train the network. And the network is trained with pixel-wise cross-entropy loss

$$\mathcal{L}_{ce} = \mathcal{L}_{ce}(\hat{D}^S; \theta) + \mathcal{L}_{ce}(D^T; \theta), \tag{12}$$

where

$$\mathcal{L}_{ce}(\hat{D}^S; \theta) = -\sum_i \sum_{(\tilde{x}, \tilde{y}) \in I_i^S} \log[P(L_i^S(\tilde{x}, \tilde{y})|\tilde{x}, \tilde{y}; I_i^S, \theta)], \tag{13}$$

$$\mathcal{L}_{ce}(D^T; \theta) = -\sum_i \sum_{(\hat{x}, \hat{y}) \in I_i^T} \log[P(L_i^T(\hat{x}, \hat{y})|\hat{x}, \hat{y}; I_i^T, \theta)]. \tag{14}$$

In Eq. (13) and Eq. (14), I_i^S and I_i^T are the i-th images in the $\hat{D}^S$ and D^T, respectively. Note that only the gradients coming from the positive source pixels $(\tilde{x}, \tilde{y}) \in I^S$ and target pixels $(\hat{x}, \hat{y})$ with reliable pseudo labels are back-propagated in each iteration.

Algorithm 1: Content-Consistent Matching

Input: parameters θ; source images D^S and labels L^S, target images from D^T
Initialize θ with source trained segmentation model.
for *m=1 to M* **do**
  Update target pseudo labels L^T for each $I^T \in D^T$ and select target pixels $(\hat{x}, \hat{y})$ with reliable pseudo labels;
  Select positive source samples $\hat{D}^S$ and their positive pixels $(\tilde{x}, \tilde{y})$
  for *n=1 to N* **do**
    1) forward and compute the $\mathcal{L}$ according to Eq. (17);
    2) back-propagating the gradients and update θ.
  end
end

3.4 Objective

Additionally, we introduce entropy regularization to regularize the adaptation

$$\mathcal{L}_{ent}(D^S;\theta) = -\sum_i \sum_c \sum_{(x,y)\in I_i^S} P(c|x,y;I_i^S,\theta)\log[P(c|x,y;I_i^S,\theta)], \tag{15}$$

$$\mathcal{L}_{ent}(D^T;\theta) = -\sum_i \sum_c \sum_{(x,y)\in I_i^T} P(c|x,y;I_i^T,\theta)\log[P(c|x,y;I_i^T,\theta)], \tag{16}$$

And the entropy regularization is imposed on all the source and target images.

In total, the objective of our training procedure is

$$\mathcal{L} = \mathcal{L}_{ce}(\hat{D}^S;\theta) + \mathcal{L}_{ce}(D^T;\theta) + \lambda(\mathcal{L}_{ent}(D^S;\theta) + \mathcal{L}_{ent}(D^T;\theta)). \tag{17}$$

where λ is a constant indicating the strength of entropy regularization.

Our algorithm is summarized in Algorithm 1. Note that we update the target pseudo labels and perform source selection every N steps of network update. We perform selection and network update in such an asynchronous way because the network update is a relatively slower process, and this way enables more efficient and effective training.

4 Experiments

4.1 Dataset and Evaluation Metric

We evaluate our methods on two popular transfer tasks, GTA5 [36] → Cityscapes [11] and SYNTHIA [37] → Cityscapes. For the source dataset, GTA5 contains 24996 images with resolution 1914 × 1052, and SYNTHIA contains 9400 images with resolution 1280 × 760. For the target, Cityscapes contains 2975 images for training and 500 images for validation with image resolution 2048 × 1024. Following the settings in [34,41,44], we train the model on the source dataset (GTA5 or

Table 1: Experiment results of GTA5 → Cityscapes.

GTA5 → CityScapes																					
	Meth.	road	side.	buil.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	motor	bike	mIoU
Source Only	–	60.6	17.4	73.9	17.6	20.6	21.9	31.7	15.3	79.8	18.1	71.1	55.2	22.8	68.1	32.3	13.8	3.4	34.1	21.2	35.7
AdaptSeg[41]	AT	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
ADVENT[44]	AT	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
CLAN[34]	AT	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
DISE[1]	AT	91.5	47.5	82.5	31.3	25.6	33.0	33.7	25.8	82.7	28.8	82.7	62.4	30.8	85.2	27.7	34.5	6.4	25.2	24.4	45.4
SWD[24]	AT	92.0	46.4	82.4	24.8	24.0	35.1	33.4	34.2	83.6	30.4	80.9	56.9	21.9	82.0	24.4	28.7	6.1	25.0	33.6	44.5
SSF-DAN[13]	AT	90.3	38.9	81.7	24.8	22.9	30.5	37.0	21.2	84.8	38.8	76.9	58.8	30.7	85.7	30.6	38.1	5.9	28.3	36.9	45.4
MaxSquare[7]	–	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	85.3	39.4	78.2	63.0	22.9	84.6	36.4	43.0	5.5	**34.7**	33.5	46.4
MRNet[56]	–	90.5	35.0	84.6	34.3	24.0	36.8	**44.1**	42.7	84.5	33.6	82.5	**63.1**	34.4	**85.8**	32.9	38.2	2.0	27.1	41.8	48.3
PyCDA[28]	ST	90.5	36.3	84.4	32.4	**28.7**	34.6	36.4	31.5	**86.8**	37.9	78.5	62.3	21.5	85.6	27.9	34.8	18.0	22.9	**49.3**	47.4
CRST[59]	ST	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1
CAG[50]	ST	90.4	51.6	83.8	34.2	27.8	**38.4**	25.3	**48.4**	85.4	38.2	78.1	58.6	**34.6**	84.7	21.9	42.7	**41.1**	29.3	37.2	50.2
SIM[46]	ST	90.1	44.7	**84.8**	34.3	**28.7**	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
BDL[56]	AS	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	**43.6**	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
Ours (CCM)	ST	**93.5**	**57.6**	84.6	**39.3**	24.1	25,2	35.0	17.3	85.0	40.6	**86.5**	58.7	28.7	**85.8**	**49.0**	**56.4**	5.4	31.9	43.2	**49.9**

Table 2: Experiment results of SYNTHIA → Cityscapes. The mIoU* denotes the mean IoU over classes without "*".

SYNTHIA → CityScapes																			
	Meth.	road	side.	buil.	wall*	fence*	pole*	light	sign	vege.	sky	pers.	rider	car	bus	motor	bike	mIoU	mIoU*
Source Only	–	47.1	23.3	75.6	7.1	0.1	23.9	5.1	9.2	74.0	73.5	51.1	20.9	39.1	17.7	18.4	34.0	34.5	40.1
AdaptSeg[41]	AT	84.3	42.7	77.5	–	–	–	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	–	46.7
ADVENT[44]	AT	85.6	42.2	79.7	–	–	–	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	–	48.0
CLAN[34]	AT	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
SSF-DAN[13]	AT	84.6	41.7	80.8	–	–	–	11.5	14.7	80.8	**85.3**	57.5	21.6	82.0	36.0	19.3	34.5	–	50.0
MaxSquare[7]	–	82.9	40.7	80.3	10.2	**0.8**	25.8	12.8	18.2	82.5	82.2	53.1	18.0	79.0	31.4	10.4	35.6	41.4	48.2
CAG[50]	ST	84.7	40.8	81.7	7.8	0.0	**35.1**	13.3	22.7	84.5	77.6	**64.2**	27.8	80.9	19.7	22.7	48.3	44.5	–
pyCDA[28]	ST	75.5	30.9	**83.3**	**20.8**	0.7	32.7	**27.3**	**33.5**	**84.7**	85.0	64.1	25.4	**85.0**	45.2	21.2	32.0	**46.7**	**53.3**
SIM[46]	ST	83.0	44.0	80.3	–	–	–	17.1	15.8	80.5	81.8	59.9	**33.1**	70.2	37.3	28.5	45.8	–	52.1
BDL[27]	AS	**86.0**	**46.7**	80.3	–	–	–	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	–	51.4
ours (CCM)	ST	79.6	36.4	80.6	13.3	0.3	25.5	22.4	14.9	81.8	77.4	56.8	25.9	80.7	**45.27**	**29.9**	**52.0**	45.2	52.9

SYNTHIA) and the training set of Cityscapes and report the result on the validation set of Cityscapes. We only transfer on the classes shared between the source domain and the target domain. For the evaluation metric, we evaluate our methods with mean Intersection over Union (mIoU).

4.2 Implementation Detail

We start from DeepLabV2-Res101 [3,17] with the backbone pretrained on the ImageNet [12]. Then we firstly finetune the whole network on the source data and use such a source-trained network to initialize the target (adaptation) model.

We choose to use Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 5×10^{-4}. The learning rate decreases following the poly policy with power at 0.9. The initial learning rate is set to 7.5×10^{-5}. The M in Algorithm 1 is set to 6 and the N is set to 2 epochs, *i.e.* we train for 6 loops where each loop contains 2 epochs. For all the transfer tasks, the hyper-parameters γ_{img}, λ, r, and K are set to 0.4, 0.4, 0.1, and 10 respectively. The γ_{pix} is set to 0.9, 0.6 for GTA5 and SYNTHIA respectively.

For image preprocessing, we resize the shorter side of images to 720 and crop a patch with resolution 600×600 randomly. Besides, horizontal flip and random

Table 3: Effect of different key components. The "CCM-SLM" stands for semantic layout matching, and "CCM-Fix" denotes source samples and pixels are only selected at the start of self-training. All the results are compared with our self-training baseline. Self-training means the network trained with cross-entropy loss and entropy regularization, without the source selection via CCM.

Module	GTA5→Cityscapes		SYNTHIA→Cityscapes	
	mIoU	Gain	mIoU	Gain
Self-training	48.1	-	41.2	-
CCM-Fix	48.9	+0.8	44.2	+3.0
CCM-SLM	48.8	+0.7	41.9	+0.7
CCM	49.9	+1.8	45.2	+4.0

scale between 0.5 and 1.5 are introduced as data augmentation. For evaluation, images from Cityscapes are resized to 1024 × 512 as input and the mIoU is calculated on predictions upsampled to 2048 × 1024.

4.3 Comparison with the state-of-the-arts

We evaluate our method on two unsupervised domain adaptation tasks: GTA5 → Cityscapes and SYNTHIA → Cityscapes. The results are presented in Table 1 and Table 2, respectively. In both tables, we use "AT" and "ST" to denote approaches established on adversarial training and self-training respectively, while "AS" indicates methods utilizing both. All the models are based on DeepLabV2-Res101 backbone, except that pyCDA [28] is based on PSPNet [54] and CAG [50] is based on DeepLabV3+ [6][4]. It can be seen that our method outperforms source only baseline with a large margin, which verifies the effectiveness of our approach.

For the task GTA5 → Cityscapes, we achieve 49.9% on mIoU, comparable to previous state-of-the-art method CAG [50] (50.2%) which is trained with a much larger resolution (*i.e.* 2200 × 1100). For the task from SYNTHIA → Cityscapes, to make a fair comparison, we report the mIoU on 13 classes (excluding "Wall", "Fence", and "Pole") and 16 classes. Our method achieves 52.9% and 45.2% mIoU on 13 classes and 16 classes respectively, both of which perform favorably against previous state-of-the-arts.

Specifically, despite its simplicity, CCM outperforms previous state-of-the-art adversarial-training (denote as "AT") based method "SSF-DAN" [13] by +4.5% and +2.9% on GTA5 → Cityscapes and SYNTHIA → Cityscapes, respectively. Compared with methods established on self-training, CCM achieves comparable or even better results. For example, our method is on par with pyCDA [28], *i.e.* 49.9% (ours) vs. 47.4% (pyCDA) on GTA5 → Cityscapes and 45.2%/52.9% (mIoU and mIoU* of ours) vs. 46.7%/53.3% (mIoU and mIoU* of pyCDA) on SYNTHIA → Cityscapes. It is worth noting that pyCDA adopts AdaBN [26] to

[4] https://github.com/RogerZhangzz/CAG_UDA/issues/6

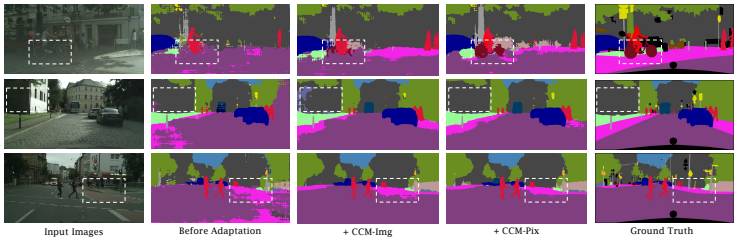


Fig. 4: Visualization of the segmentation results (GTA5 → Cityscapes). Pay attention to the dashed box to see the effect of different modules.

enhance its adaptation performance, which can also be employed in our framework to improve the performance further. Also, our method mainly focuses on selecting positive source information to mitigate the domain shift and help the adaptation, which is also complementary to these methods and can be combined with them to boost the adaptation performance.

4.4 Ablation Studies

Effect of different key components We verify the effect of each key component in our framework in Table 3. It can be seen that compared to the source-only results, self-training improves the adaptation performance apparently. Despite such a strong baseline, "CCM-SLM", which selects positive source samples through proposed SLM, improves beyond self-training by +0.7% on both tasks. Further, through combining SLM with pixel similarity matching, "CCM" improves beyond self-training by +1.8% and +4.0% for the GTA5 → Cityscapes and SYNTHIA → Cityscapes, respectively. These noticeable performance gains verify the effectiveness of the proposed matching and selection strategy.

Fig. 4 gives an intuitive illustration about the effect of CCM. It can be seen that through adaptation with SLM, the pixel-level predictions have been largely improved. Further, pixel-wise similarity matching enables the adapted model to learn more details about the object and thus leads to more accurate predictions.

Compared with the "CCM-Fix" which only selects positive source samples and their positive pixels at the start of self-training and adopts them throughout the adaptation, actively update the positive source set (denoted as "CCM") achieves noticeable improvement (*i.e.* +1.0% for both tasks). This is because source samples may contribute differently to the adaptation at different training stages and the matching results should be updated as the target predictions evolve, The results imply that self-training and CCM could benefit each other and cooperate to mitigate the domain shift.

Visualization of semantic layout matching results In Fig. 5, we show the source images retrieved by individual target images via semantic layout matching at the final training stage. In each row, (c-e) are source images retrieved by the

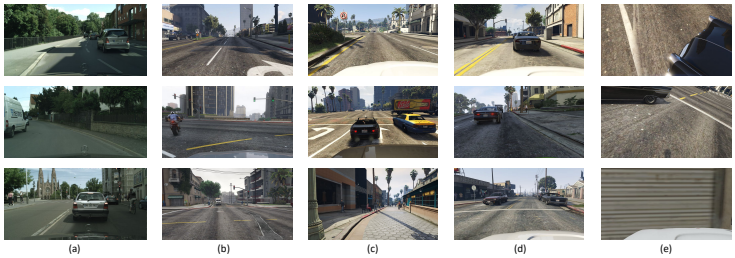


Fig. 5: Examples retrieved by semantic layout matrix(SLM). In each row, (b-e) are source images retrieved by target sample (a), where (b-d) are positive samples and (e) are negative ones.

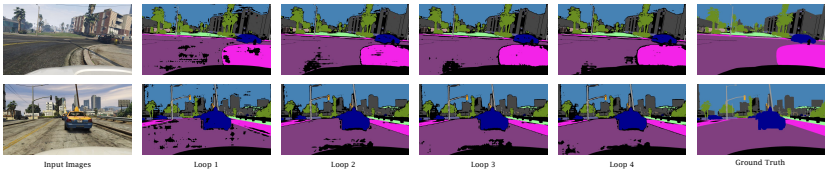


Fig. 6: Visualization of selected source pixels at different training stages. As the training goes on, the ignored source pixels become more and more concentrated on the object boundary. The black area denotes the ignored pixels during training. The results are based on task GTA5 → Cityscapes.

target sample (a) via SLM, in which (b-d) are top positive samples and (e) are negative ones with the lowest matching scores.

It can be seen that similar layout is shared among the target samples and the retrieved source samples. For example, all positive source samples in the first row have trees on the left. Moreover, it is also obvious that negative source samples on the right-most column (e) have totally different layout. Additionally, the retrieved source samples remain reasonable variations in appearances, which will also benefit the generalization of adapted model. All of these results give an intuitive illustration why semantic layout matching can help reduce the domain shift and improve the generalization ability.

Effect of pixel-wise similarity matching Fig. 6 demonstrates the selected source pixels through pixel-wise similarity matching during the training. It can be seen that as the training goes on, the ignored pixels become more and more concentrated on the object boundary, which is reasonable and implies that the adaptation keeps improving. Moreover, at the early stage, we notice that the ignored pixels are ambiguous ones that are hard to distinguish, *e.g.* the pixels of the cracks on the road. The pixel selection through pixel-wise similarity matching enables the model to learn in a curriculum way to an extent.

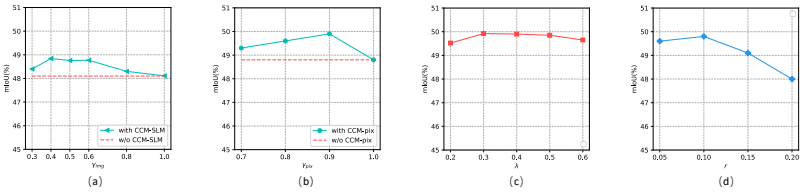


Fig. 7: (a): Performance with/without CCM-SLM under different γ_{img}. (b): Performance with/without CCM-Pix under different γ_{pix}. (c): Sensitivity analysis of λ. (d): Sensitivity analysis of r. The results shown are based on the task GTA5 $\rightarrow$ Cityscapes. The trend on another task is similar.

Sensitivity to the hyper-parameters We investigate the sensitivity of our method to the hyper-parameters γ_{img}, γ_{pix}, λ, r, and show the results in Fig. 7. From Fig. 7 (a), it can be seen that trained with SLM, with the increase of γ_{img}, the mIoU firstly increases then decreases, illustrating a bell shape curve. The mIoU decreases when γ_{img} is above a certain threshold, indicating that there exist negative samples harming the adaptation and it is necessary to perform source sample selection to exclude such negative samples. With SLM, the optimal mIoU achieved is 0.7% higher than that trained without SLM. The trend of sensitivity to γ_{pix} is similar to that of γ_{img}. Our method achieves consistent improvement over baselines (the red lines) within a wide range of γ_{img} and γ_{pix}.

Table 4: Sensitivity to K

K	5	10	15
mIoU(%)	49.8	49.9	49.7

As illustrated in Fig. 7 (c), entropy regularization provides consistent improvement within a wide range of λ. From Fig. 7 (d), we observe that our method is also robust to r within a wide range. When r is above a certain threshold, the performance drops because more inaccurate target data is involved in training. Besides, we analyze the sensitivity of our model to K and report the results in Table 4, which further verifies the robustness of our approach.

5 Conclusion

In this paper, we propose using Content-Consistent Matching (CCM), which consists of Semantic Layout Matching and Pixel-wise Similarity Matching, to match and select positive source data to facilitate the adaptive training of the segmentation model. Our matching strategy is performed from both the image-level and the pixel-level, *i.e.* semantic layout matching selects the positive source samples, and pixel-wise similarity matching emphasizes the effective source pixels. Experiment results on two representative benchmarks demonstrate that our method performs favorably against previous state-of-the-arts.

Acknowledgement This work is in part supported by ARC DECRA DE190-101315 and ARC DP200100938.

References

1. Chang, W.L., Wang, H.P., Peng, W.H., Chiu, W.C.: All about structure: Adapting structural information across domains for boosting semantic segmentation. In: IEEE CVPR (June 2019) 10
2. Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J.: Progressive feature alignment for unsupervised domain adaptation. In: IEEE CVPR (June 2019) 2, 4
3. 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. IEEE TPAMI **40**(4), 834–848 (2017) 10
4. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR **abs/1706.05587** (2017), http://arxiv.org/abs/1706.05587 2
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (September 2018) 2
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. pp. 801–818 (2018) 11
7. Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: IEEE ICCV (October 2019) 10
8. Chen, Y.C., Lin, Y.Y., Yang, M.H., Huang, J.B.: Crdoco: Pixel-level domain transfer with cross-domain consistency. In: IEEE CVPR (June 2019) 4
9. Cheng, B., Chen, L.C., Wei, Y., Zhu, Y., Huang, Z., Xiong, J., Huang, T.S., Hwu, W.M., Shi, H.: Spgnet: Semantic prediction guidance for scene parsing. In: IEEE ICCV. pp. 5218–5228 (2019) 2
10. Choi, J., Kim, T., Kim, C.: Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In: IEEE ICCV (October 2019) 4
11. 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: IEEE CVPR (2016) 9
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009) 10
13. 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: IEEE ICCV (October 2019) 2, 4, 10, 11
14. Feng, Q., Kang, G., Fan, H., Yang, Y.: Attract or distract: Exploit the margin of open set. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7990–7999 (2019) 3
15. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML. p. 1180–1189. ICML'15, JMLR.org (2015) 4
16. Gong, R., Li, W., Chen, Y., Gool, L.V.: Dlow: Domain flow for adaptation and generalization. In: IEEE CVPR (June 2019) 3, 4
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 10
18. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017) 2, 4

19. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV (2018) 3
20. Huang, Z., Wang, X., Wei, Y., Huang, L., Shi, H., Liu, W., Huang, T.: Ccnet: Criss-cross attention for semantic segmentation. TPAMI (2020) 2
21. Jiang, L., Meng, D., Zhao, Q., Shan, S., Hauptmann, A.G.: Self-paced curriculum learning. In: AAAI (2015) 4
22. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR. pp. 4893–4902 (2019) 4
23. Kang, G., Zheng, L., Yan, Y., Yang, Y.: Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In: ECCV (September 2018) 3
24. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: IEEE CVPR (June 2019) 2, 4, 10
25. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: ICML. vol. 3, p. 2 (2013) 4
26. Li, Y., Wang, N., Shi, J., Hou, X., Liu, J.: Adaptive batch normalization for practical domain adaptation. Pattern Recognit. **80**, 109–117 (2018). https://doi.org/10.1016/j.patcog.2018.03.005, https://doi.org/10.1016/j.patcog.2018.03.005 11
27. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: IEEE CVPR (June 2019) 2, 4, 10
28. Lian, Q., Lv, F., Duan, L., Gong, B.: Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In: IEEE ICCV (October 2019) 2, 4, 10, 11
29. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: IEEE CVPR (Jul 2017) 2
30. Liu, X., Li, S., Kong, L., Xie, W., Jia, P., You, J., Kumar, B.: Feature-level frankenstein: Eliminating variations for discriminative recognition. In: IEEE CVPR (June 2019) 4
31. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. pp. 3431–3440 (2015) 2, 3
32. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. p. 97–105. ICML'15, JMLR.org (2015) 3
33. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791 (2015) 3
34. 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: IEEE CVPR (2019) 2, 4, 9, 10
35. Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596 (2014) 4
36. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV. LNCS, vol. 9906, pp. 102–118. Springer International Publishing (2016) 9
37. 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: IEEE CVPR (June 2016) 9
38. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: ICML. pp. 2988–2997. JMLR. org (2017) 4
39. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV. pp. 443–450. Springer (2016) 3

40. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: IEEE CVPR (2018) 4
41. 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: IEEE CVPR (2018) 2, 4, 9, 10
42. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: IEEE CVPR (July 2017) 4
43. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. CoRR **abs/1412.3474** (2014), http://arxiv.org/abs/1412.3474 3
44. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: IEEE CVPR (2019) 2, 4, 9, 10
45. Wang, Z., Yu, M., Wei, Y., Feris, R., Xiong, J., Hwu, W.m., Huang, T.S., Shi, H.: Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In: CVPR. pp. 12635–12644 (2020) 4
46. Wang, Z., Yu, M., Wei, Y., Feris, R., Xiong, J., Hwu, W.m., Huang, T.S., Shi, H.: Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In: IEEE CVPR (June 2020) 10
47. Wu, J., Huang, Z., Acharya, D., Li, W., Thoma, J., Paudel, D.P., Gool, L.V.: Sliced wasserstein generative models. In: IEEE CVPR (June 2019) 4
48. Wu, Z., Han, X., Lin, Y.L., Gokhan Uzunbas, M., Goldstein, T., Nam Lim, S., Davis, L.S.: Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In: ECCV (September 2018) 4
49. Xie, S., Zheng, Z., Chen, L., Chen, C.: Learning semantic representations for unsupervised domain adaptation. In: Dy, J., Krause, A. (eds.) ICML. Proceedings of Machine Learning Research, vol. 80, pp. 5423–5432. PMLR, Stockholmsmässan, Stockholm Sweden (10–15 Jul 2018), http://proceedings.mlr.press/v80/xie18c.html 4
50. Zhang, Q., Zhang, J., Liu, W., Tao, D.: Category anchor-guided unsupervised domain adaptation for semantic segmentation. In: NeuralPS. pp. 433–443 (2019) 2, 4, 10, 11
51. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaptation. In: IEEE CVPR (June 2018) 4
52. Zhang, Y., David, P., Foroosh, H., Gong, B.: A curriculum domain adaptation approach to the semantic segmentation of urban scenes. IEEE TPAMI pp. 1–1 (2019). https://doi.org/10.1109/TPAMI.2019.2903401 4
53. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: IEEE ICCV. p. 6 (Oct 2017) 4
54. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE CVPR (2017) 2, 11
55. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. arXiv preprint arXiv:2003.03773 (2020) 4
56. Zheng, Z., Yang, Y.: Unsupervised scene adaptation with memory regularization in vivo. IJCAI (2020) 10
57. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: IEEE ICCV (2017) 3
58. Zou, Y., Yu, Z., Kumar, B.V., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV. pp. 289–305 (2018) 2, 4

59. Zou, Y., Yu, Z., Liu, X., Kumar, B.V., Wang, J.: Confidence regularized self-training. In: IEEE ICCV (October 2019) 2, 4, 10