Class-Balanced Pixel-Level Self-Labeling for Domain Adaptive Semantic Segmentation

Ruihuang Li[1], Shuai Li[1], Chenhang He[1], Yabin Zhang[1], Xu Jia[2], Lei Zhang[1*]
[1]The Hong Kong Polytechnic University, [2]Dalian University of Technology
{csrhli, csshuaili, csche, csybzhang, cslzhang}@comp.polyu.edu.hk, xjia@dlut.edu.cn

Abstract

Domain adaptive semantic segmentation aims to learn a model with the supervision of source domain data, and produce satisfactory dense predictions on unlabeled target domain. One popular solution to this challenging task is self-training, which selects high-scoring predictions on target samples as pseudo labels for training. However, the produced pseudo labels often contain much noise because the model is biased to source domain as well as majority categories. To address the above issues, we propose to directly explore the intrinsic pixel distributions of target domain data, instead of heavily relying on the source domain. Specifically, we simultaneously cluster pixels and rectify pseudo labels with the obtained cluster assignments. This process is done in an online fashion so that pseudo labels could co-evolve with the segmentation model without extra training rounds. To overcome the class imbalance problem on long-tailed categories, we employ a distribution alignment technique to enforce the marginal class distribution of cluster assignments to be close to that of pseudo labels. The proposed method, namely Class-balanced Pixel-level Self-Labeling (CPSL), improves the segmentation performance on target domain over state-of-the-arts by a large margin, especially on long-tailed categories. The source code is available at `https://github.com/lslrh/CPSL`.

1. Introduction

Semantic segmentation is a fundamental computer vision task, which aims to make dense semantic-level predictions on images [8,27,28,43,53]. It is a key step in numerous applications, including autonomous driving, human-machine interaction, and augmented reality, to name a few. In the past few years, the rapid development of deep Convolutional Neural Networks (CNNs) has boosted semantic segmentation significantly in terms of accuracy and efficiency. However, the performance of deep models trained in one domain often drops largely when they are applied to unseen domains. For example, in autonomous driving the segmentation model is confronted with great challenges when weather conditions are changing constantly [56]. A natural way to improve the generalization ability of segmentation model is to collect data from as many scenarios as possible. However, it is very costly to annotate pixel-wise labels for a large amount of images [11]. More effective and practical approaches are required to address the domain shifts of semantic segmentation.

Unsupervised Domain Adaptation (UDA) provides an important way to transfer the knowledge learned from one labeled source domain to another unlabeled target domain. For example, we can collect many synthetic data whose dense annotations are easy to get by using game engines such as GTA5 [36] and SYNTHIA [37]. Then the question turns to how to adapt the model trained from a labeled synthetic domain to an unlabeled real image domain. Most previous works of UDA bridge the domain gap by aligning data distributions at the image level [17,25,33], feature level [7,17,18,24] or output level [29,32,39], through adversarial training or auxiliary style transfer networks. However, these techniques will increase the model complexity and make the training process unstable, which impedes their reproducibility and robustness.

Another important approach is self-training [52,56,57], which alternatively generates pseudo labels by selecting high-scoring predictions on target domain and provides supervision for the next round of training. Though these methods have produced promising performance, there are still some major limitations. On one hand, the segmentation model tends to be biased to source domain so that the pseudo labels produced on target domain are error-prone; on the other hand, highly-confident predictions may only provide very limited supervision information for the model training. To solve these issues, some methods [50,51] have been proposed to produce more accurate and informative pseudo labels. For example, instead of using the classifier trained on source domain to generate pseudo labels, Zhang *et al*. [51] assigned pseudo labels to pixels based on

*Corresponding Author

their distances to the category prototypes. These prototypes, however, were built in source domain and usually deviated much from the target domain. ProDA [50] leveraged the feature distances from prototypes to perform online rectification, but it was challenging to construct prototypes for long-tailed categories, which often led to unsatisfactory performance.

Different from previous self-training methods which use classifier-based noisy pseudo labels for supervision, in this paper we propose to perform online pixel-level self-labeling via clustering on target domain, and use the resulting soft cluster assignments to correct pseudo labels. Our idea comes from the fact that pixel-wise cluster assignments could reveal the intrinsic distributions of pixels in target domain, and provide useful supervision for model training. Compared to conventional label generation methods that are often biased towards source domain, cluster assignment in target domain is more reliable as it explores inherent data distribution. Considering that the classes of segmentation dataset are highly imbalanced (please refer to Fig. 2), we employ a distribution alignment technique to enforce the class distribution of cluster assignments to be close to that of pseudo labels, which is more favorable to class-imbalanced dense prediction tasks. The proposed Class-balanced Pixel-level Self-Labeling (CPSL) module works in a plug-and-play fashion, which could be seamlessly incorporated into existing self-training framework for UDA. The major contributions of this work are summarized as follows:

- A pixel-level self-labeling module is developed for domain adaptive semantic segmentation. We cluster pixels in an online fashion and simultaneously rectify pseudo labels based on the resulting cluster assignments.
- A distribution alignment technique is introduced to align the class distribution of cluster assignments to that of pseudo labels, aiming to improve the performance over long-tailed categories. A class-balanced sampling strategy is adopted to avoid the dominance of majority categories in pseudo label generation.
- Extensive experiments demonstrate that the proposed CPSL module improves the segmentation performance on target domain over state-of-the-arts by a large margin. It especially shows outstanding results on long-tailed classes such as "motorbike", "train", "light", *etc*.

2. Related Work

Semantic Segmentation. The goal of semantic segmentation is to segment an image into regions of different semantic categories. While the Fully Convolutional Networks (FCNs) [28] have greatly boosted the performance of semantic segmentation, they have relatively small receptive field to explore visual context. Many later works focus on how to enlarge the receptive field of FCNs to model long-range context dependencies of images, such as dilated convolution [8], multi-layer feature fusion [27], spatial pyramid pooling [53] and variants of non-local blocks [15, 20, 22]. However, directly applying these models to unseen domains will induce poor segmentation performance because of their weak generalization ability. Therefore, many domain adaptation techniques have been proposed to improve model generalization ability on new domains.

Domain Adaptation for Semantic Segmentation. Recently, many works have been proposed to bridge the domain gap and improve the adaptation performance. The most representative ones are adversarial training-based methods [19, 23, 34, 39, 40], which aim to align different domains on intermediate features or network predictions. Style transfer-based methods [6, 9, 10, 44, 48] minimize domain gap at the image level. For example, Chang *et al*. [6] proposed to disentangle an image into domain-invariant structures and domain-specific textures for image translation. The training process of these models is rather complex since multiple networks, such as discriminators or style transfer networks, have to be trained concurrently.

Another important technique for UDA is self-training [24, 26, 32, 51, 55, 57], which iteratively generates pseudo labels on target data for model update. Zou *et al*. [55] proposed a class-balanced self-training method for domain adaption of semantic segmentation. To reduce the noise in pseudo labels, Zou *et al*. [57] further proposed a confidence regularized self-training method, which treated pseudo labels as trainable latent variables. Lian *et al*. [26] constructed a pyramid curriculum for exploring various properties about the target domain. Zhang *et al*. [51] enforced category-aware feature alignment by choosing the prototypes of source domain as guided anchors. ProDA [50] went further by employing the feature distances from each pixel to prototypes to correct pseudo labels pre-computed by the source model. These methods, however, neglect either the pixel-wise intrinsic structures or inherent class distribution of target domain images, tending to be biased to source domain or majority classes.

Clustering-based Representation Learning. Our work is also related to clustering-based methods [2–4, 21, 45–47, 54, 58]. Caron *et al*. [4] iteratively performed k-means on latent representations and used the produced cluster assignments to update network parameters. Recently, Asano *et al*. [3] cast the cluster assignment problem as an optimal transport problem which can be solved efficiently through a fast variant of the Sinkhorn-Knopp algorithm. SwAV [5] performed clustering while enforcing consistency among the cluster assignments of different augmentations of the same image. In this paper, we extend self-labeling from image-level classification to pixel-level semantic segmentation. In addition, different from Asano *et al*. [3] and Caron *et al*. [4], we com-

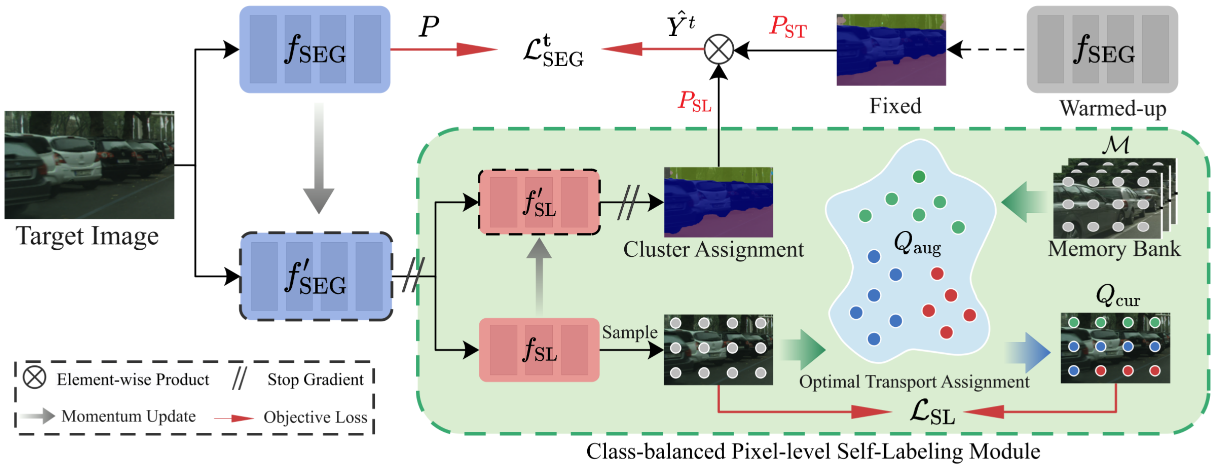

Figure 1. The framework of Class-balanced Pixel-level Self-Labeling (CPSL). The model contains a main segmentation network f_{SEG} and its momentum-updated version f'_{SEG}. The f'_{SEG} is followed by a self-labeling head f_{SL} and its momentum version f'_{SL}, which projects pixel-wise feature embedding into a class probability vector. The pixel-level self-labeling module produces soft cluster assignment P_{SL} to gradually rectify soft pseudo label P_{ST}. Then the segmentation loss $\mathcal{L}^t_{\text{SEG}}$ is computed between the prediction map P and the rectified pseudo label $\hat{Y}^t$. To train the self-labeling head, we randomly sample pixels from each image, and use the memory bank $\mathcal{M}$, which contains previous batches of pixel features, to augment the current batch. Then we compute the optimal transport assignment Q_{aug} over the augmented data by enforcing class balance, and use the assignment of current batch Q_{cur} to compute the self-labeling loss $\mathcal{L}_{\text{SL}}$.

pute cluster assignments in an online fashion, making our method scalable to dense pixel-wise prediction tasks.

3. Method

3.1. Overall Framework

In the setting of unsupervised domain adaptation for semantic segmentation, we are provided with a set of labeled data in source domain $\mathcal{D}_S = \{(X^s_n, Y^s_n)\}^{N_S}_{n=1}$, where X^s_n is the source image with label Y^s_n and N_S is the number of images, as well as a set of N_T unlabeled images X^t_n in target domain $\mathcal{D}_T = \{X^t_n\}^{N_T}_{n=1}$. Both domains share the same C classes. Our goal is to learn a model by using the labeled source data in $\mathcal{D}_S$ and unlabeled target data in $\mathcal{D}_T$, which could perform well on unseen test data in the target domain.

The overall framework of our proposed CPSL is shown in Fig. 1. We propose a pixel-level self-labeling module (highlighted in the green color box) to explore the intrinsic pixel-wise distributions of the target domain data via clustering, and to reduce the noise in pseudo labels. Before the training, we first generate a soft pseudo label map $P_{\text{ST}} \in \mathbb{R}^{H\times W\times C}$ for each target domain image by a warmed-up model that is pre-trained on the source domain data. The obtained P_{ST} is usually error-prone because of the large domain shift. Therefore, in the training process, we rectify P_{ST} incrementally with the soft cluster assignment, denoted by $P_{\text{SL}} \in \mathbb{R}^{H\times W\times C}$. Specifically, the rectification of P_{ST} is conducted as follows:

$$\hat{Y}^{t,(c)}_{n,i} = \begin{cases} 1, & \text{if } c = \underset{c*}{argmax}(P^{(c*)}_{\text{SL},n,i} \cdot P^{(c*)}_{\text{ST},n,i}) \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where $\hat{Y}^{t,(c)}_{n,i}$ denotes the c-th element of rectified pseudo label at the i-th pixel of target image X^t_n. $P^{(c*)}_{\text{SL},n,i}$ represents the probability that the i-th pixel of X^t_n belongs to the $c*$-th category. Eq. 1 has a similar formulation to [35, 38, 50], where P_{SL} can be regarded as the weight map to modulate the softmax probability map P_{ST}. The cluster assignment P_{SL} exploits the inherent data distribution of target domain, thus it is highly complementary to the classifier-based pseudo label P_{ST} which heavily relies on source domain.

We define the segmentation loss on target domain, denoted by $\mathcal{L}^t_{\text{SEG}}$, as the pixel-level cross-entropy loss between the segmentation probability map $P_n \in \mathbb{R}^{H\times W\times C}$ and the rectified pseudo label $\hat{Y}^t_n$ of target image X^t_n:

$$\mathcal{L}^t_{\text{SEG}} = -\sum_{n=1}^{N_T}\sum_{i=1}^{H\times W}\sum_{c=1}^{C} \hat{Y}^{t,(c)}_{n,i} \log P^{(c)}_{n,i}. \quad (2)$$

In addition, the loss on source domain, denoted by $\mathcal{L}^s_{\text{SEG}}$, can be defined as the standard pixel-wise cross-entropy on the labeled images:

$$\mathcal{L}^s_{\text{SEG}} = -\sum_{n=1}^{N_S}\sum_{i=1}^{H\times W}\sum_{c=1}^{C} Y^{s,(c)}_{n,i} \log P^{(c)}_{n,i}. \quad (3)$$

Then the total segmentation loss $\mathcal{L}_{\text{SEG}}$ is obtained as the sum of them: $\mathcal{L}_{\text{SEG}} = \mathcal{L}^t_{\text{SEG}} + \mathcal{L}^s_{\text{SEG}}$.

In the following subsections, we will explain in detail the design of our CPSL module.

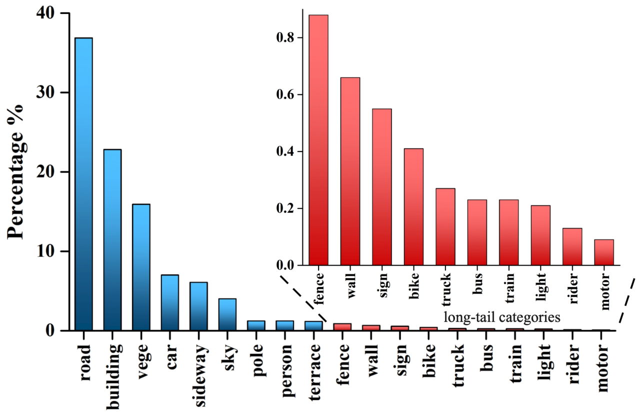

Figure 2. The class distribution of the Cityscapes dataset.

3.2. Online Pixel-Level Self-Labeling

Pixel-Level Self-Labeling. Conventional self-training based methods usually use a model pre-trained on source domain to produce pseudo labels, which often contain much noise [51,55,57]. To clean the pseudo labels, we propose to perform pixel-level self-labeling via clustering on target domain and use the obtained cluster assignments to rectify the pseudo labels. The basic motivation is that pixel-wise clustering could reveal the intrinsic structures of target domain data, and it is complementary to the classifier trained on source domain data. Thus, cluster assignments could provide extra supervision for training a domain adaptive segmentation model.

Specifically, we first extract features from an input image to obtain $Z \in \mathbb{R}^{H\times W\times D}$ and normalize it with $z_i = \frac{z_i}{||z_i||_2}$, where z_i is the i-th feature vector of Z with length D. Then we randomly sample a group of pixels $\hat{Z} = [z_1, \cdots, z_M]$ from each image, and pass them through a self-labeling head f_{SL}. Finally, we obtain their class probability vectors $\hat{P} = [p_1, \cdots, p_M]$ by taking a softmax operation:

$$p_m^{(c)} = \frac{\exp(\frac{1}{\tau} f_{\mathrm{SL}}^{(c)}(z_m))}{\sum_{c'} \exp(\frac{1}{\tau} f_{\mathrm{SL}}^{(c')}(z_m))}, \; c \in \{1, \cdots, C\}, \tag{4}$$

where $f_{\mathrm{SL}}^{(c)}(z_m)$ is the c-th element of the output of z_m from self-labeling head. $p_m^{(c)}$ denotes the probability that the m-th pixel belongs to the c-th category. τ is a temperature parameter. Considering there is no ground truth label available for target data, we train the head f_{SL} through a self-labeling mechanism [3] with the following objective function:

$$\begin{aligned} &\mathcal{L}_{\mathrm{SL}} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{c=1}^{C} q_m^{(c)} \log p_m^{(c)} \quad s.t.\, Q \in \mathbb{Q}, \\ &\text{with} \quad \mathbb{Q} := \{Q \in \mathbb{R}_+^{C\times M} | Q\mathbf{1}_M = r, Q^T\mathbf{1}_C = h\}. \end{aligned} \tag{5}$$

The above formula is an instance of the optimal transport problem [13], where $Q = \frac{1}{M}[q_1, \cdots, q_M]$ is a transport assignment and it is restricted to be a probability matrix by satisfying the constraint $\mathbb{Q}$. $\mathbf{1}_C$ and $\mathbf{1}_M$ denote the vectors of ones with dimension C and M, respectively. r and h are the marginal projections of Q onto its rows and columns, respectively.

By formulating the cluster assignment problem as an optimal transport problem, the optimization of Eq. 5 with respect to variable Q can be solved efficiently by the iterative Sinkhorn-Knopp algorithm [13]. The optimal solution is obtained by:

$$Q^* = \mathrm{diag}(\alpha)\exp(\frac{f_{\mathrm{SL}}(\hat{Z})}{\varepsilon})\,\mathrm{diag}(\beta), \tag{6}$$

where $\alpha \in \mathbb{R}^C$ and $\beta \in \mathbb{R}^M$ are two renormalization vectors which can be computed efficiently in linear time even for dense prediction tasks. ε is a temperature parameter.

Then by fixing label assignment Q, the self-labeling head f_{SL} is updated by minimizing $\mathcal{L}_{\mathrm{SL}}$ with respect to $\hat{P}$, which is the same as training with cross-entropy loss.

Weight Initialization. We use the soft cluster assignment P_{SL} to rectify the classifier-based pseudo label P_{ST}. However, the clustering categories usually mismatch those of the classifier, resulting in performance degradation. To overcome this issue, we initialize the weight of self-labeling head f_{SL} with category prototypes. Specifically, we compute the prototypes $[\bar{\mathbf{z}}_1, \cdots, \bar{\mathbf{z}}_C]$ for each category through:

$$\bar{\mathbf{z}}_c = \frac{1}{|\Gamma_c|}\sum_{n=1}^{N_T}\sum_{i=1}^{H\times W} Y_{\mathrm{ST},n,i}^{(c)} \cdot z_{n,i}, \tag{7}$$

where $|\Gamma_c|$ denotes the number of pixels belonging to the c-th category in all images. Y_{ST} is the hard version of P_{ST}. Then the self-labeling process can be regarded as assigning pixels to different prototypes. In this way, the clustering categories are able to match classification categories.

Online Cluster Assignment. Different from Asano *et al.* [3], where the assignment Q is computed over the full dataset, we conduct online clustering on data batches during training. Considering that the number of samples in a mini-batch is often too small to cover all categories, and the class distribution varies largely across different batches, we augment the features $\hat{Z}$ with a memory bank $\mathcal{M}$, which is updated on-the-fly, to reduce the randomness of sampling. Specifically, throughout the training process, we maintain a queue of 65,536 pixel features from previous batches in $\mathcal{M}$. In each iteration, we compute the optimal transport assignment on the augmented data Z_{aug}, denoted by Q_{aug}, but only the assignment of current batch, denoted by Q_{cur}, is used to compute the self-labeling loss $\mathcal{L}_{\mathrm{SL}}$. In this way, we could alternatively update the self-labeling head f_{SL} and use it to generate more accurate cluster assignment P_{SL} online. Hence, the pseudo labels will be improved incrementally by the resulting cluster assignments, and the noise will be gradually reduced without extra rounds of training.

3.3. Class-Balanced Self-Labeling

As shown in Fig. 2, there exists severe class-imbalance in current semantic segmentation datasets. Some long-tailed classes have very limited pixels (*e.g.*, "traffic light", "sign"), and some classes only appear in a few images (*e.g.*, "motorbike", "train"). Such a problem will make it difficult to train a robust segmentation model, especially for those long-tailed classes. In this work, we propose two techniques to address this issue, *i.e.*, class-balanced sampling and distribution alignment.

Class-Balanced Sampling. We randomly sample pixels from each image, which makes the class distribution of data in memory bank $\mathcal{M}$ approach to that of the whole dataset. In order to make sure that the pixels of long-tailed categories can be selected equally, we sample from different categories with the same proportion, *i.e.*, $\frac{M}{H \times W}$, where M is the number of pixels to be sampled in each image. For each input image X_n^t, we first compute its class distribution δ_n through

$$\delta_n^{(c)} = \frac{1}{H \times W} \sum_{i}^{H \times W} \hat{Y}_{n,i}^{t,(c)}, \tag{8}$$

where $\delta_n^{(c)}$ denotes the proportion of pixels belonging to the c-th category in image X_n^t. Then the number of samples M_c for each category c is decided by:

$$M_c = \left\lfloor M \times \delta_n^{(c)} \right\rfloor. \tag{9}$$

If image X_n^t does not contain certain classes of pixels, we will randomly sample the rest pixels from other categories to make up M samples.

Distribution Alignment. As discussed in [3, 4], simultaneously optimizing Q and $\hat{P}$ in Eq. 5 may lead to degenerated results that all data points are trivially assigned to a single cluster. To avoid this, Asano *et al.* [3] constrained that Q should induce an equipartition of the data. However, this constraint is not reasonable and it will degrade the performance if the ground truth class distribution of the data, denoted by δ_{gt}, is not uniform. In the Cityscapes dataset [11], for example, the number of pixels of the largest category ("road") is approximately 300 times that of the smallest category ("motorbike").

To overcome this problem, we propose a novel technique, namely distribution alignment, to align the distribution of cluster assignments to ground truth class distribution δ_{gt}, aiming at partitioning pixels into subsets of unequal sizes. However, δ_{gt} is unknown since the true labels of target domain data are unavailable. Thus we propose to employ the moving average of pseudo labels' class distribution δ_{pseudo} to approximate δ_{gt}. Specifically, we first initialize δ_{pseudo} based on the fixed pseudo labels Y_{ST}^t as follows:

$$\delta_{pseudo}^{(c)}|_0 = \frac{1}{N_T \times H \times W} \sum_{n}^{N_T} \sum_{i}^{H \times W} Y_{\mathrm{ST},n,i}^{t,(c)}. \tag{10}$$

Over the course of training, we compute the class distribution δ_n of each image through Eq. 8. Then the class distribution δ_{pseudo} after each training iteration k is updated with a momentum $\alpha \in [0, 1]$:

$$\delta_{pseudo}^{(c)}|_k = \alpha \delta_{pseudo}^{(c)}|_{k-1} + (1 - \alpha) \delta_n^{(c)}. \tag{11}$$

Finally, we enforce the class distribution of cluster assignments, denoted by r in Eq. 5, to be close to δ_{pseudo}:

$$r = \delta_{pseudo}, \quad h = \frac{1}{M} \mathbf{1}_M. \tag{12}$$

Our empirical results (please refer to Fig. 6) demonstrate that the proposed distribution alignment technique effectively avoids the dominance of majority classes during training. Please refer to Sec. 4.3 for more discussions.

3.4. Loss Function

As shown in Fig. 1, we employ momentum encoder to stabilize the self-labeling process. To further improve the model generalization ability on target domain and alleviate the bias inherited from source domain, following [1, 50], we impose consistency regularization on the segmentation network. Specifically, we generate a weakly-augmented image X_w and a strongly-augmented image X_s from the same input image X, and pass X_w through the momentum segmentation network f'_{SEG} to generate a probability map P_w, which is used to supervise the output P_s of strongly-augmented image X_s from f_{SEG}. Then we enforce P_w and P_s to be consistent via:

$$\mathcal{L}_{\mathrm{REG}} = \sum_{n=1}^{N_T} \sum_{i=1}^{H \times W} \left(\ell_{\mathrm{KL}} \left(P_{w,n,i}, P_{s,n,i} \right) + \ell_{\mathrm{KL}} \left(P_{s,n,i}, P_{w,n,i} \right) \right), \tag{13}$$

where ℓ_{KL} denotes the KL-divergence. $P_{s,n,i}$ and $P_{w,n,i}$ represent the i-th pixel of the segmentation probability maps P_s and P_w of image X_n, respectively.

The overall loss function is defined as:

$$\mathcal{L}_{\mathrm{TOTAL}} = \mathcal{L}_{\mathrm{SEG}} + \lambda_1 \mathcal{L}_{\mathrm{SL}} + \lambda_2 \mathcal{L}_{\mathrm{REG}}, \tag{14}$$

where λ_1 and λ_2 are trade-off parameters. $\mathcal{L}_{\mathrm{SL}}$ and $\mathcal{L}_{\mathrm{REG}}$ are complementary to each other. The former uses pixel-level cluster assignment P_{SL} to rectify the pseudo label P_{ST}, which effectively dilutes the bias to source domain, while the latter improves model generalization ability by applying data augmentations on inputs and consistency regularization on outputs.

Method	road	sideway	building	wall	fence	pole	light	sign	vege	terrace	sky	person	rider	car	truck	bus	train	motor	bike	mIoU
AdaptSeg [39]	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
CyCADA [17]	86.7	35.6	80.1	19.8	17.5	38.0	39.9	41.5	82.7	27.9	73.6	64.9	19.0	65.0	12.0	28.6	4.5	31.1	42.0	42.7
ADVENT [41]	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
CBST [56]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
FADA [42]	92.5	47.5	**85.1**	37.6	32.8	33.4	33.8	18.4	85.3	37.7	**83.5**	63.2	**39.7**	87.5	32.9	47.8	1.6	34.9	39.5	49.2
CAG_UDA [51]	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
FDA [48]	92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
PIT [30]	87.5	43.4	78.8	31.2	30.2	36.3	39.3	42.0	79.2	37.1	79.3	65.4	37.5	83.2	**46.0**	45.6	25.7	23.5	49.9	50.6
IAST [31]	**93.8**	**57.8**	**85.1**	39.5	26.7	26.2	43.1	34.7	84.9	32.9	88.0	62.6	29.0	87.3	39.2	49.6	23.2	34.7	39.6	51.5
ProDA [50]	91.5	52.4	82.9	42.0	**35.7**	40.0	44.4	43.3	**87.0**	**43.8**	79.5	66.5	31.4	86.7	41.1	52.5	0.0	45.4	**53.8**	53.7
CPSL (ours)	91.7	52.9	83.6	**43.0**	32.3	**43.7**	**51.3**	42.8	85.4	37.6	81.1	**69.5**	30.0	**88.1**	44.1	**59.9**	24.9	**47.2**	48.4	**55.7**
ProDA+*distill*	87.8	56.0	79.7	**46.3**	**44.8**	45.6	53.5	53.5	**88.6**	**45.2**	82.1	70.7	**39.2**	88.8	45.5	59.4	1.0	48.9	**56.4**	57.5
CPSL+*distill*	**92.3**	**59.9**	**84.9**	45.7	29.7	**52.8**	**61.5**	**59.5**	87.9	41.5	**85.0**	**73.0**	35.5	**90.4**	**48.7**	**73.9**	**26.3**	**53.8**	53.9	**60.8**

Table 1. Experimental results on the GTA5 → Cityscapes adaptation task. The top score is highlighted in **bold** font.

Method	road	sideway	building	wall	fence	pole	light	sign	vege	sky	person	rider	car	bus	motor	bike	mIoU^{13}	mIoU^{16}
AdaptSeg [39]	79.2	37.2	78.8	-	-	-	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	45.9	-
ADVENT [41]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	48.0	41.2
CBST [56]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	48.9	42.6
CAG_UDA [51]	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	51.5	44.5
PIT [30]	83.1	27.6	81.5	8.9	0.3	21.8	26.4	33.8	76.4	78.8	64.2	27.6	79.6	31.2	31.0	31.3	51.8	44.0
FADA [42]	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	52.5	45.2
FDA [48]	79.3	35.0	73.2	-	-	-	19.9	24.0	61.7	82.6	61.4	**31.1**	83.9	40.8	38.4	51.1	52.5	-
PyCDA [26]	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	53.3	46.7
IAST [31]	81.9	41.5	83.3	17.7	**4.6**	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	33.1	52.7	57.0	49.8
SAC [1]	**89.3**	**47.2**	**85.5**	26.5	1.3	**43.0**	45.5	32.0	**87.1**	**89.3**	63.6	25.4	86.9	35.6	30.4	53.0	59.3	52.6
ProDA [50]	87.1	44.0	83.2	**26.9**	0.7	42.0	45.8	**34.2**	86.7	81.3	68.4	22.1	87.7	50.0	31.4	38.6	58.5	51.9
CPSL (ours)	87.3	44.4	83.8	25.0	0.4	42.9	**47.5**	32.4	86.5	83.3	**69.6**	29.1	**89.4**	**52.1**	**42.6**	**54.1**	**61.7**	**54.4**
ProDA+*distill*	**87.8**	**45.7**	84.6	**37.1**	**0.6**	44.0	54.6	37.0	**88.1**	84.4	74.2	24.3	88.2	**51.1**	40.5	45.6	62.0	55.5
CPSL+*distill*	87.2	43.9	**85.5**	33.6	0.3	**47.7**	**57.4**	**37.2**	87.8	**88.5**	**79.0**	**32.0**	**90.6**	49.4	**50.8**	**59.8**	**65.3**	**57.9**

Table 2. Experimental results on the SYNTHIA → Cityscapes adaptation task. The top score is highlighted in **bold** font.

4. Experiments

4.1. Experimental Settings

Implementation Details. We implement the segmentation model with DeepLabv2 [8] and employ ResNet-101 [16] as the backbone, which is pre-trained on ImageNet. The segmentation model is warmed up by applying adversarial training like [39]. The input images are randomly cropped to 896×512, and the batch size is set as 4. We employ a series of data augmentations such as RandAugment [12], Cutout [14], CutMix [49], and add photometric noise, including color jitter, random blur, *etc*. SGDM is used as the optimizer. The initial learning rate of segmentation model and self-labeling head are set to 10^{-4} and 5×10^{-4}, which decay exponentially with power 0.9. The weight decay and momentum are set to 2×10^{-4} and 0.9, respectively. The trade-off parameters λ_1, λ_2 and the temperature parameters τ, ε are empirically set to 0.1, 5, 0.08, and 0.05, respectively. The length of memory bank is set to $65,536$ and we sample 512 pixels per image for clustering ($M = 512$), that is, there are 128 images in the memory bank. For the momentum networks, the momentum is set to 0.999. Our model is trained with four Tesla V100 GPUs on PyTorch.

Datasets. Following [31, 51, 52], we adopt two synthetic datasets (GTA5 [36], SYNTHIA [37]) and one real dataset (Cityscapes [11]) in the experiments. The GTA5 dataset contains 24,966 images with resolution 1914×1052. The corresponding dense annotations are generated by game engine. The SYNTHIA dataset contains 9,400 images of 1280×760 pixels and it has 16 common categories with Cityscapes, which contains 2,975 training images and 500 validation images of resolution 2048×1024.

4.2. Comparisons with State-of-the-Arts

We name the proposed method as Class-balanced Pixel-level Self-Labeling (CPSL). Following [50], after the training converges, we also conduct two more knowledge distillation rounds to transfer the knowledge to a student model pre-trained in a self-supervised manner, and the resulting model is called "CPSL+distill". We compare our models with representative and state-of-the-art methods, which can be categorized to two main groups: adversarial training-based methods, including AdaptSeg [39], CyCADA [17], FADA [42], ADVENT [41], and self-training based methods, including CBST [55], IAST [31], CAG_UDA [51], ProDA [50], SAC [1]. Following previous works, the results on validation set are reported in terms of category-wise Intersection over Union (IoU) and mean IoU (mIoU).

GTA5→Cityscapes. The results on GTA5→Cityscapes task are reported in Tab. 1. Our CPSL achieves the best IoU score on 7 out of 19 categories, and it achieves the highest mIoU score, outperforming the second best method

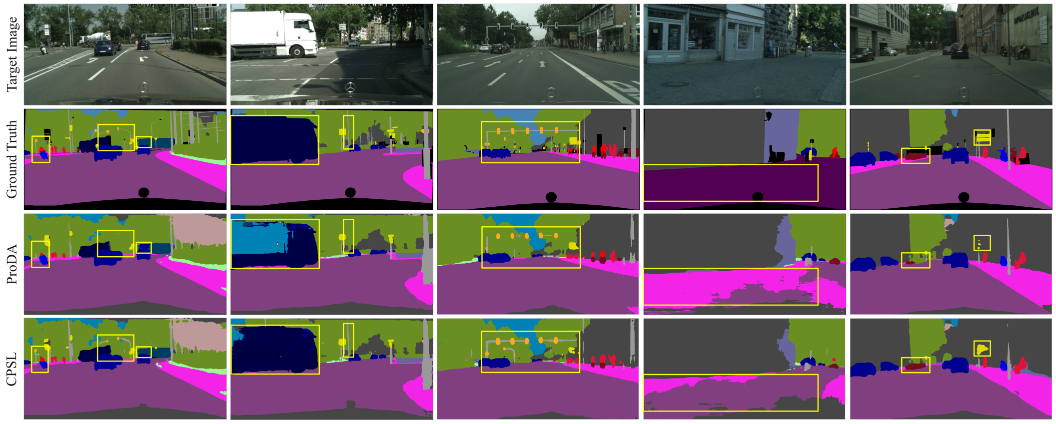

Figure 3. Qualitative results of our method and ProDA [50] on the GTA5→Cityscapes task.

Configuration	mIoU	Δ
w/o SL	47.8	-7.9
w/o CB	51.8	-3.9
w/o ST	39.4	-16.3
w/o Init	49.9	-5.8
w/o Aug	54.2	-1.5
w/o Mom	54.6	-1.1
CPSL	**55.7**	-

Table 3. Ablation studies on the key components of our proposed method.

# samples	mIoU
64	54.9
128	55.3
256	55.5
512	**55.7**
1024	54.3
2048	53.4

Table 4. The influence of the number of samples per image on performance.

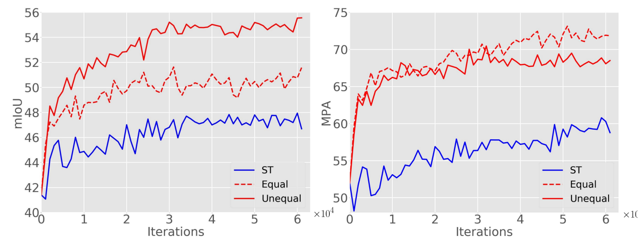

Figure 4. The mIoU and mean pixel accuracy (MPA) scores evaluated on the validation set with equal/unequal partition constraint.

ProDA [50] by a large margin of 2.0. This can be attributed to the exploration of inherent data distribution of target domain, which provides extra supervision for training. By applying knowledge distillation, there is a further performance gain of 5.1, achieving 60.8 mIoU, which is by far the new state-of-the-art. It is worth mentioning that our method performs especially well on long-tailed categories, such as "pole", "light", "train", and "motor". For example, ProDA fails on the small class "train" due to the difficulties in constructing prototypes for long-tailed categories. By applying distribution alignment, CPSL alleviates the class-imbalance problem, attaining 24.9 IoU on "train" without sacrificing the performance on other categories.

SYNTHIA→Cityscapes. This adaptation task is more challenging than the previous one because of the large domain gap. The mIoUs over 13 classes (mIoU^{13}) and 16 classes (mIoU^{16}) are reported in Tab. 2. Our model still achieves significant improvements over competing methods on this task. Specifically, CPSL achieves the mIoU of 54.4 and 61.7 over 16 and 13 categories, surpassing the second best method SAC [1] by 1.8 and 2.4, respectively. This owes to the fact that CPSL reduces the label noise and calibrates the bias to source domain. The results are further improved to 57.9 and 65.3 in terms of mIoU after distillation. Among all the 16 categories, our method tops over six of them, especially on the hardest categories, such as "light", "motorbike", "bike", and so on.

Qualitative Results. Fig. 3 shows the qualitative segmentation results of our method and ProDA [50] on GTA5→Cityscapes task. As can be seen, our method improves the performance on long-tailed classes substantially, *e.g.* "pole", "light", "bus", thanks to the class-balanced sampling and distribution alignment techniques. ProDA [50] does not perform well on these categories since it does not explicitly enforce class balance in training.

4.3. Discussions

Ablation Study. We conduct ablation studies on the GTA5→Cityscapes task to investigate the role of each component in CPSL. For the convenience of expression, we abbreviate 'self-labeling', 'self-training', 'class balance', 'weight initialization', 'data augmentation', and 'momentum encoder' with 'SL', 'ST', 'CB', 'Init', 'Aug', 'Mom'. Tab. 3 shows the corresponding results by switching off each component. We have the following observations.

First, removing the SL component leads to a drop of 7.9 in mIoU, while disabling CB component leads to a drop of 3.9 in mIoU. This demonstrates they play key roles in improving the segmentation performance by exploring the intrinsic data structures of target domain images. Second, training without the pseudo labels produced by ST causes a significant drop of 16.3 in mIoU. This is not surprising because simultaneously updating network parameters and generating pseudo labels will lead to a degenerate solution [50, 51]. Third, randomly initializing the self-labeling head (w/o Init) results in a decline of 5.8 in mIoU, which is attributed to the mismatch between clustering and classifi-

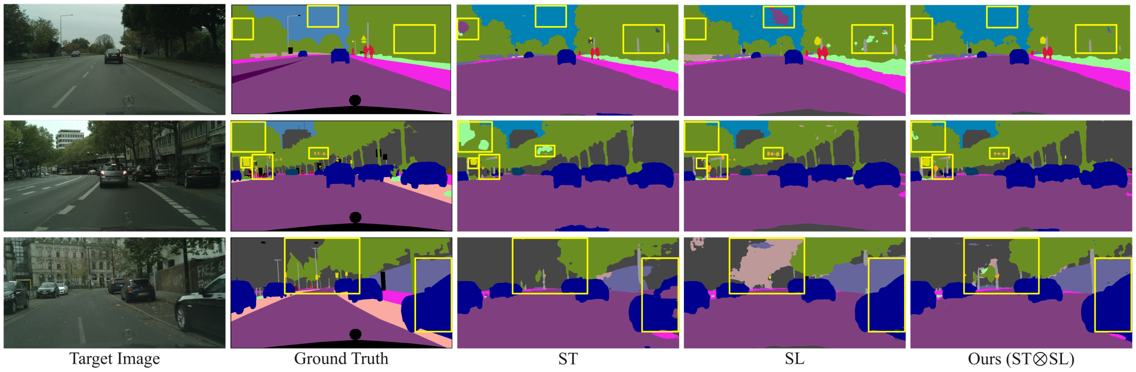

Figure 5. The complementarity between label assignments produced by self-training (ST) and self-labeling (SL).

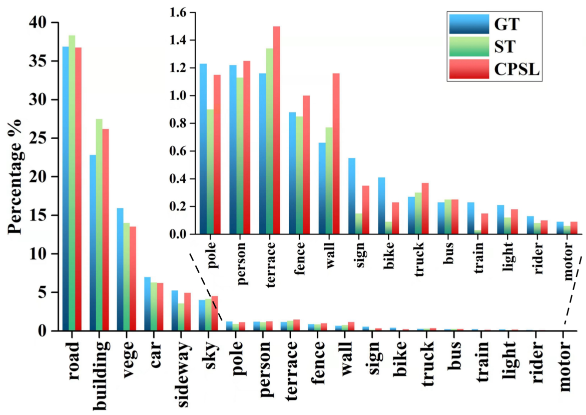

Figure 6. The comparisons of class distributions on Cityscapes dataset. 'GT' denotes the ground truth class distribution. 'ST' and 'CPSL' denote the class distributions of pseudo labels produced by self-training and class-balanced pixel-level self-labeling.

cation categories. Fourth, Aug and Mom components bring an improvement of 1.7 and 1.1 in mIoU.

Unequal Partition Constraint. To further analyze the effect of unequal partition on class-imbalanced dataset, we plot the curves of mIoU and MPA scores with different partition constraints in Fig. 4, where a huge gap can be observed in terms of mIoU. However, equal partition slightly outperforms unequal partition in terms of MPA. This is not surprising because many pixels belonging to large categories are assigned to small categories under the equal partition constraint, largely improving pixel accuracy of small classes without influencing much large classes. Thus the MPA score is improved. More details can be found in the *supplemental files*.

Self-Training (ST) vs. Self-Labeling (SL). We explore the complementarity of label assignments produced by ST and SL, and visualize the results in Fig. 5. One can draw a conclusion that the integration of ST and SL in our CPSL leads to better results than any one of them. Specifically, ST performs better on large categories which are easy to transfer, such as "sky" and "building", while SL has advantages on small categories such as "light" and "pole". Therefore, the pixels that are wrongly classified in one view will be corrected in another view.

The Effect of Distribution Alignment. We compare the class distributions of labels produced by CPSL and conventional self-training (ST). As illustrated in Fig. 6, the results of ST mismatch heavily to ground truth (GT). Its predictions are biased towards majority categories, *e.g*. 'road' and 'building', ignoring small categories such as 'train', 'sign' and 'bike'. CPSL calibrates the bias and produces a class distribution closer to GT. This demonstrates that CPSL can capture the inherent class distribution of target domain and avoids gradual dominance of majority classes.

Parameter Sensitivity Analysis. In Tab. 4, we evaluate the segmentation performance on GTA5→Cityscapes task with different number of samples per image. Our method is robust to this parameter within a wide range. More analyses can be found in *supplemental materials*.

Limitation. Although the proposed CPSL alleviates the bias to source domain with the self-labeling assignment, it still relies on the self-training based pseudo labels, which may lead to confirmation bias. We consider to develop a fully clustering-based assignment method in future works.

5. Conclusion

We proposed a plug-and-play module, namely Class-balanced Pixel-level Self-Labeling (CPSL), which could be seamlessly incorporated into self-training pipelines to improve the domain adaptive semantic segmentation performance. Specifically, we conducted pixel-level clustering online and used the resulting cluster assignments to rectify pseudo labels. On one hand, the label noise was reduced and the bias to source domain was calibrated by exploring pixel-level intrinsic structures of target domain images. On the other hand, CPSL captured inherent class distribution of target domain, which effectively avoided gradual dominance of majority classes. Both the qualitative and quantitative analyses demonstrated that CPSL outperformed the existing state-of-the-arts by a large margin. In particular, it achieved great performance gains on long-tailed classes without sacrificing the performance on other categories.

References

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15384–15394, 2021. 5, 6, 7

[2] Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. *Advances in Neural Information Processing Systems*, 33:4660–4671, 2020. 2

[3] Y M Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR 2020 : Eighth International Conference on Learning Representations*, 2020. 2, 4, 5

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2, 5

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9912–9924, 2020. 2

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

[7] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 627–636, 2019. 1

[8] 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 transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1, 2, 6

[9] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1791–1800, 2019. 2

[10] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6830–6840, 2019. 2

[11] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1, 5, 6

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 6

[13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*, volume 26, pages 2292–2300, 2013. 4

[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 6

[15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1989–1998, 2017. 1, 6

[18] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 1

[19] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018. 2

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[21] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *International Conference on Machine Learning*, pages 2849–2858, 2019. 2

[22] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 2

[23] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12975–12984, 2020. 2

[24] Ruihuang Li, Xu Jia, Jianzhong He, Shuaijun Chen, and Qinghua Hu. T-svdnet: Exploring high-order prototypical correlations for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9991–10000, 2021. 1, 2

[25] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In

2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6936–6945, 2019. 1

[26] Qing Lian, Lixin Duan, Fengmao Lv, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6757–6766, 2019. 2, 6

[27] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 1, 2

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 2

[29] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2507–2516, 2019. 1

[30] Fengmao Lv, Tao Liang, Xiang Chen, and Guosheng Lin. Cross-domain semantic segmentation via domain-invariant interactive relation transfer. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4334–4343, 2020. 6

[31] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 415–430, 2020. 6

[32] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12435–12445, 2021. 1, 2

[33] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018. 1

[34] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3764–3773, 2020. 2

[35] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018. 3

[36] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 1, 6

[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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1, 6

[38] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4077–4087, 2017. 3

[39] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018. 1, 2, 6

[40] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1456–1465, 2019. 2

[41] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2517–2526, 2019. 6

[42] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV (14)*, pages 642–659, 2020. 6

[43] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021. 1

[44] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gökhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018. 2

[45] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 478–487, 2016. 2

[46] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6509–6518, 2020. 2

[47] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5147–5156, 2016. 2

[48] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 2, 6

[49] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regular-

ization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 6

[50] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. 1, 2, 3, 5, 6, 7

[51] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *arXiv preprint arXiv:1910.13049*, 2019. 1, 2, 4, 6, 7

[52] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2039–2049, 2017. 1, 6

[53] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 2

[54] Chengxu Zhuang, Alex Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6002–6012, 2019. 2

[55] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018. 2, 4, 6

[56] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 297–313, 2018. 1, 6

[57] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5982–5991, 2019. 1, 2, 4

[58] Miguel Ángel Bautista, Artsiom Sanakoyeu, Ekaterina Tikhoncheva, and Björn Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 3846–3854, 2016. 2