Prototypical Contrast Adaptation for Domain Adaptive Semantic Segmentation

Zhengkai Jiang[1], Yuxi Li[1], Ceyuan Yang[2], Peng Gao[2], Yabiao Wang[1†], Ying Tai[1], and Chengjie Wang[1†]

[1] Tencent Youtu Lab
[2] The Chinese University of Hong Kong
{zhengkjiang, caseywang}@tencent.com

Abstract. Unsupervised Domain Adaptation (UDA) aims to adapt the model trained on the labeled source domain to an unlabeled target domain. In this paper, we present Prototypical Contrast Adaptation (ProCA), a simple and efficient contrastive learning method for unsupervised domain adaptive semantic segmentation. Previous domain adaptation methods merely consider the alignment of the intra-class representational distributions across various domains, while the inter-class structural relationship is insufficiently explored, resulting in the aligned representations on the target domain might not be as easily discriminated as done on the source domain anymore. Instead, ProCA incorporates inter-class information into class-wise prototypes, and adopts the class-centered distribution alignment for adaptation. By considering the same class prototypes as positives and other class prototypes as negatives to achieve class-centered distribution alignment, ProCA achieves state-of-the-art performance on classical domain adaptation tasks, *i.e., GTA5 → Cityscapes and SYNTHIA → Cityscapes*. Code is available at ProCA.

Keywords: Domain Adaptive Semantic Segmentation, Prototypical Contrast Adaptation

1 Introduction

Semantic segmentation is a fundamental computer vision task, which requires per-pixel predictions for a given image. Recently, with the development of deep neural networks (DNN) [11,14,16–18,44,46], semantic segmentation has achieved remarkable progress [2,24,49]. However, state-of-the-art methods still suffer from significant performance drops when the distribution of testing data is different from training data owing to the domain shifts problem [27, 30, 32]. At the same time, labeling pixel-wise large-scale semantic segmentation in the target domain is time-consuming and prohibitively expensive. Thus, Unsupervised Domain Adaptation (UDA) is a promising direction to solve such problem by adapting a model trained from largely labeled source domain to an unlabeled target domain without additional cost of annotations.

†Corresponding author.

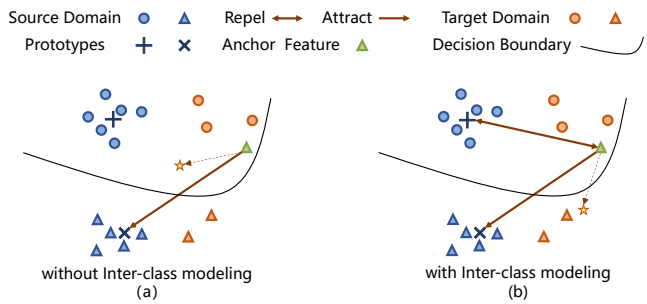

Fig. 1. Illustration of inter-class modeling. $\star$ means the adapted feature of target domain. With explicit inter-class constraints during adaptation, adapted features of target domain can appear at the right place of decision boundary.

Several works relying on adversarial training [12, 40, 42] have achieved remarkable progress for UDA semantic segmentation. These methods reduce the domain discrepancy between source and target domains by minimizing a series of adversarial training losses. Specifically, it is formulated as a two-player game, where a backbone network (*i.e. ResNet-101 backbone*) serves as the feature extractor, while a discriminator identifies which domain the features are derived from. To reach equilibrium in this minmax game, it requires the backbone network to produce the domain invariant representations for generalization. Such adversarial training will result in aligned and indistinguishable feature distributions between two domains. However, even though the global feature distributions across domains become closer, it is not guaranteed that pixels attributing to different semantic categories in the target domain are well separated, leading to poor generalization ability and even inferior performance.

To tackle the issues above, some works attempt to take the category-wise information into account. The idea of encouraging high-confidence predictions is achieved by minimizing the entropy of the output [42]. The discrepancies between the outputs of two classifiers are utilized to achieve category-level alignment implicitly [27]. In addition, a fine-grained adversarial learning framework [43] is proposed to incorporate class information into domain discrimination, which helps to align features at a fine-grained level. However, prior approaches tend to apply such adversarial training in the intra-class, without considering the consistency of the representational structure between the source and target domains. Namely, to some extent, multiple categories on the target domain could be projected to a same group, which are usually well-discriminated on the source domain on the contrary. Therefore, merely considering the intra-class distributional alignment might be insufficient to make the best of the learned representations from labeled source data.

In order to fully exploit the class-level information, we propose *Prototypical Contrast Adaptation* (ProCA) for unsupervised domain adaptive semantic segmentation. Intuitively, the same category on different domains is supposed

to share the high representational similarity. Therefore, multiple prototypes, *i.e.*, the approximated representational centroid of various categories are utilized to depict the inter-class relationship for both source and target domains. Specifically, after acquiring the segmentation model trained only on the source domain, category-wise prototypes features are obtained by calculating the centroids of features on the source domain. Then, contrastive learning is introduced into domain adaptation process. In particular, a pixel on the target domain is pulled closer to its corresponding prototype with the same class as its estimated pseudo-label and pushed away from other prototypes. In addition, in order to be invariant to domains, category-wise prototypes would be further updated by the current features of two domains. Besides, such prototypical contrastive adaptation scheme is applied at the feature and output level simultaneously. Based on the self-training framework, we further improve the performance with class-aware pseudo-label thresholds.

Experimental results on the domain adaptation benchmarks for semantic segmentation, *i.e., GTA5 → Cityscapes and SYNTHIA → Cityscapes* further demonstrate the effectiveness of our approach, leading to the state-of-the-art performance. Specifically, with the DeepLab-v2 networks and ResNet-101 backbone, we achieve Cityscapes [4] semantic segmentation mIoU by 56.3% and 53.0% when adapting from GTA5 [35] and SYNTHIA [36] datasets, largely outperforming previous state-of-the-arts.

We summarize the major contributions as follows:

- We propose *Prototypical Contrastive Adaptation* (*ProCA*) by explicitly introducing constraints on features of different categories for UDA problem in semantic segmentation. This is implemented by not only pulling closer to prototypes with the same class, but also pushing away from prototypes with different classes simultaneously. A multi-level variant is also designed to further improve the adaptation ability.
- Online prototypes updating scheme is introduced to improve the domain invariance and discriminant ability of class-wise prototypes.
- Combined with self-training method of class-wise adaptive thresholds, the proposed method achieves 56.3% and 52.6% mIoU when adapting GTA5 and SYNTHIA to Cityscapes, respectively, which outperforms previous state-of-the-arts by a large margin.

2 Related Works

2.1 Semantic Segmentation

Semantic segmentation is a fundamental computer vision task, which requires per-pixel predictions for a given image. Recently, with the help of convolution neural networks [24], semantic segmentation has achieved remarkable progress. Numerous approaches focus to enlarge receptive fields [2] and capture context information [49]. These methods generally require dense pixel-wise annotation

datasets, such as Cityscapes [4], PASCAL VOC [6] and ADE20K [51]. Since per-pixel level annotation of large amounts of data is time-consuming and expensive, some synthetic datasets are proposed such as GTA5 [35] and SYNTHIA [36] to generate largely labeled segmentation datasets at lower cost. However, when testing models trained on the synthetic datasets on the real-world datasets, significant performance drops are observed even for state-of-the-art methods. In presence of the domain shifts, we deal with the semantic segmentation task that aims to learn a well performing model on the target domain with only the source domain supervision.

2.2 UDA for Semantic Segmentation

Existing approaches for UDA of semantic segmentation can be primarily divided into three groups, including style transfer [31], feature alignment [8,12,13,52] and self-training [1,56]. Motivated by the recent progress of unpaired image-to-image translation works [55], researches on style transfer aim to learn the mapping from virtual to realistic data [12, 31]. Previous works on feature alignment minimize the discrepancy between source and target domains to obtain domain-invariant features. This can be achieved by directly minimizing the Maximum Mean Discrepancy (MMD) distances across domains over domain-specific layers [25] or using discriminator to train the model in an adversarial way to avoid generating domain-aware discriminative features [13]. There are also some works attempting to absorb class-wise information into feature alignment. The fine-grained adversarial learning framework [43] is proposed to incorporate class information into the discriminator, which helps to align feature in a class-aware manner, resulting better feature adaptation and performance. Approaches on self-training mainly focus on assigning pseudo-labels on target domain. Iterative self-training method is proposed [56] by alternatively generating pseudo-labels and retraining the model with a sampling module to deal with the category imbalanced issue. Uncertainty estimation [50] is proposed to rectify pseudo-label generation. Consistency based methods [1] have been adopted by enforcing consistency between predictions of different perturbations. In the work of [48], a prototype-based sample-wise pseudo-label correction scheme is proposed and embeded into a complicated multi-stage training framework to enhance segmentation performance. Nevertheless, the methods above neglect the explicit modeling of the relationship between clusters of different categories, on the contrary, we directly explore such constraints of different category centroids by prototypical contrastive adaptation. In this way, the categories with similar distributions on the target domain can be easier to distinguish, leading to superior performance.

2.3 Contrastive Learning

Contrastive learning [3,10,53] has lead remarkable performance in self-supervised representation learning. STC [17] uses contrastive learning to learn association embeddings for video instance segmentation task. For UDA semantic segmentation, CLST [29] attempts to leverage contrastive learning to learn finer adapted

feature representation. The concurrent work SDCA [21] proposes using high-order semantic information to conduct contrast adaptation for UDA segmentation, which we found that it is not necessary. In this paper, with the aid of contrastive learning, we explicitly model the relationships of pixel-wise features between different categories and domains to obtain domain-invariant representation for unsupervised domain adaptive semantic segmentation.

3 Methodology

By minimizing the distributional distance between the source and target domains, previous approaches aim to obtain the domain-invariant representations for domain adaptation problem. However, the inter-class structural relationship is insufficiently explored. As shown in Figure 1 (a), after alignment within the intra-class across two domains, it could be much more challenging to distinguish different categories since the decision boundaries identified on source domain could hardly be maintained on the target domain. Therefore, we propose a novel *category-aware prototypical contrast adaptation* which introduces multiple prototypes to explicitly model the intra-class and inter-class relationships in a contrastive manner.

Akin to previous state-of-the-art approaches [21,43,48], a segmentation model is first trained on source domain in the supervised manner. Meanwhile, multiple prototypes are initialized to represent each category. Contrast adaptation is then adopted to constrain the inter-class relationship. Besides, prototypes are updated on both source domain and target domain to enhance the domain-invariant representations. As last, we present a modified pseudo-label generation method with class-aware adaptive thresholds for self-training, leading to new state-of-the-art performances.

3.1 Preliminaries

Given the labeled source domain images $\mathcal{D}_s = \{(\mathbf{x}_n^s, y_n^s)\}_{n=1}^{N_s}$, as well as unlabeled target images $\mathcal{D}_t = \{(\mathbf{x}_n^t)\}_{n=1}^{N_t}$, the goal of UDA of semantic segmentation is to train a model on $\mathcal{D}_s$ and $\mathcal{D}_t$; and evaluate the performance on the target domain. The segmentation model consists of a feature extractor $\mathcal{F}$ and a classifier $\mathcal{C}$, which predicts pixel-wise predictions for a given image.

Following previous works [12, 21, 43], the segmentation model is first trained on the labeled source domain in a supervised manner by minimizing the loss between the prediction p_n^s and the ground-truth label $Y_n^s \in \mathbb{L}^{H\times W}, \mathbb{L} = \{1, 2, \cdots, C\}$ annotated with C category labels, for a given image $x_n^s \in \mathbb{R}^{H\times W}$. We use the standard cross-entropy loss, which can be formulated as:

$$\mathcal{L}_n^{ce} = -\sum_{n=1}^{N_s}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{C} y_{n,i,j,c}^s \log(p_{n,i,j,c}^s), \tag{1}$$

where N_s is the number of source domain images, H and W denote the height and the width of an image, i, j are the pixel index of height and width, C is the

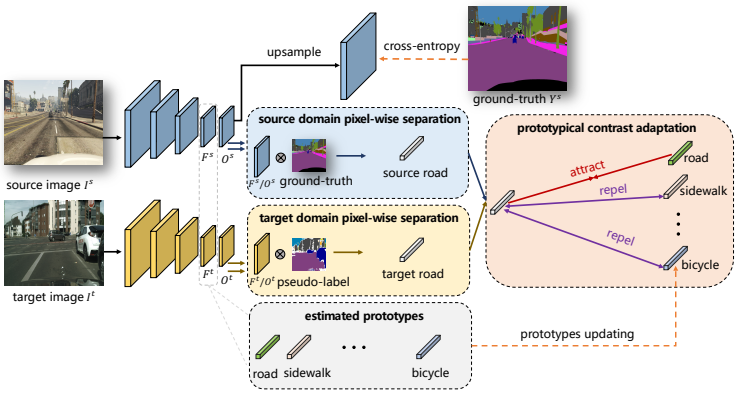

Fig. 2. The framework of proposed ProCA. For given source image I^s and target image I^t, features F^s and F^t of two domains are first obtained through a shared feature encoder $\mathcal{F}$. Then, outputs O^s and O^t are obtained by a shared classifier $\mathcal{C}$. After obtaining initialized prototypes, a pixel from two domains acts as a contrastive manner with class-aware prototypes to directly model inter-class constraints. We conduct such prototypical contrast adaption on both feature-level and output-level. At last, the initialized prototypes are also updated during training to enhance the domain-invariant representational ability.

number of categories. $p_n^s \in \mathbb{R}^{H\times W\times C}$ is the predicted probability of the image x_n^s, which is obtained by up-sampling the prediction $\mathcal{C}(\mathcal{F}(x_n^s))$. $y_n^s \in \{0,1\}^{H\times W\times C}$ is the one-hot representation of the ground-truth label Y_n^s.

3.2 Prototypical Contrast Adaptation

Here, intra-class and inter-class relations are simultaneously considered by prototypes-based contrastive learning as shown in Figure 2. Specifically, *ProCA* contains three stages, including *prototypes initialization*, *contrast adaptation* and *prototypes updating*.

Prototypes Initialization. After obtaining the model trained on the labeled source domain, the initialized class-aware prototypes can be calculated as:

$$\mathbf{p}_c^{feat} = \frac{\sum_{n=1}^{N_s}\sum_{i=1}^{H}\sum_{j=1}^{W} F_{n,i,j}^s \mathbb{1}[Y_{n,i,j}^s = c]}{\sum_{n=1}^{N_s}\sum_{i=1}^{H}\sum_{j=1}^{W} \mathbb{1}[Y_{n,i,j}^s = c]}, \tag{2}$$

where $F_{n,i,j}^s \in \mathbb{R}^d$ is the extracted source feature vector with dimension d, c is the index of categories number C, H and W denote the height and width of the features, $\mathbb{1}[Y_{n,i,j}^s = c]$ is an indicator function, which equals to 1 if $Y_{n,i,j}^s = c$ and 0 otherwise. Prototypes could be regarded as the approximated representational centroid of various categories.

Contrast Adaptation. Given an image of target domain, the corresponding feature F_n^t is extracted by the shared backbone network $\mathcal{F}$. Accordingly, its

pseudo-label $\widetilde{y}_n^t \in \{0,1\}^{H\times W\times C}$ could be produced by the classifier $\mathcal{C}$ trained on source domain. Here, pseudo-label could bridge the extracted features and their corresponding prototypes. Therefore, we could compute the similarity between features and each of prototypes, leading to a vector $P_{n,i,j}^{t\to s} = [P_{n,i,j,1}^{t\to s}, \ldots, P_{n,i,j,C}^{t\to s}]$:

$$P_{n,i,j,c}^{t\to s} = \frac{\exp(\mathbf{p}_c^{feat} \cdot F_{n,i,j}^t/\tau)}{\sum_{c=1}^{C}\exp(\mathbf{p}_c^{feat} \cdot F_{n,i,j}^t/\tau)}, \tag{3}$$

where τ is the temperature. Then, we minimize the cross entropy loss between $P_{n,i,j}^{t\to s}$ and pseudo-label $\widetilde{y}_n^t$ as:

$$\mathcal{L}_n^{t\to s} = -\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{C}\widetilde{y}_{n,i,j,c}^t \log P_{n,i,j,c}^{s\to t}. \tag{4}$$

The goal of such objective is to enforce the pixels belonging to the same category are supposed to share high representational similarity. In addition to the cross-domain adaptation, we also use source-source contrastive loss $\mathcal{L}_n^{s\to s}$ similarly:

$$\mathcal{L}_n^{s\to s} = -\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{C} y_{n,i,j,c}^s \log P_{n,i,j,c}^{s\to s}, \tag{5}$$

where y_n^s is the ground-truth one-hot source domain label, $P_{n,i,j}^{s\to s}$ is calculated similarly as Equation 3. The final pixel-prototypes contrastive loss on feature-level is:

$$\mathcal{L}_{\text{ContraFeat}} = \sum_{n=1}^{N_t}\mathcal{L}_n^{t\to s} + \sum_{n=1}^{N_s}\mathcal{L}_n^{s\to s}. \tag{6}$$

Prototypes Updating. To enhance the domain-invariant representational ability of prototypes, we propose two schemes of prototype updating along with training to incorporate target-related information into prototypes. One is to update according to the computation of strict statistical mean of global data as:

$$\mathbf{p}_c^{feat} \leftarrow \frac{\mathbf{p}_c^{feat} n_c^{feat} + \widetilde{\mathbf{p}}_c^{feat}\widetilde{n}_c^{feat}}{n_c^{feat} + \widetilde{n}_c^{feat}}, \tag{7}$$

where n_c^{feat} represents the accumulated number of pixels belonging to category c until the last update, $\widetilde{\mathbf{p}}_c^{feat}$ represents the online estimated prototypes for category c, and $\widetilde{n}_c^{feat}$ represents the total number of pixels belonging to category c from a newly appended mini-batch during training.

In addition to source domain class-wise prototypes, we also leverage target features to update prototypes during feature adaptation process. This mixed prototypes scheme could be regarded as a bridge across two domains, which could naturally interact with each other. Thus, we further propose an alternative and more stable and robust way to directly update prototypes with a mixed domain scheme:

$$\mathbf{p}_c^{feat} \leftarrow m\mathbf{p}_c^{feat^s} + (1-m)\mathbf{p}_c^{feat^t}, \tag{8}$$

where m is a hyper-parameter, which defines a constant rate of source and target prototypes updating during training. $\mathbf{p}_c^{feat^s}$ is the estimated source prototype, and $\mathbf{p}_c^{feat^t}$ is the estimated target prototype.

Label Space Adaptation. As we mentioned before, prototypes are initialized, calculated and updated at the feature level *i.e.*, output of the backbone network $\mathcal{F}$. Apart from this, we could also apply the proposed prototypical contrast adaptation in the label space *i.e.*, the output of the classifier $\mathcal{C}$. The major difference is that the dimension of prototypes becomes the number of categories rather than the hidden channels in the feature space. Accordingly, the overall prototypical contrast adaptation losses becomes:

$$\mathcal{L}_{\text{Contra}} = \mathcal{L}_{\text{ContraFeat}} + \mathcal{L}_{\text{ContraOut}}. \tag{9}$$

3.3 Combining ProCA with Self-Training

Since the proposed category-aware prototypical contrast adaptation is orthogonal to self-training based methods, we further improve the adaptation performance through the self-training strategy following previous works [21, 30].

Class-wise Adaptive Pseudo-Label Thresholds. After the prototypical contrast adaptation stage, we could obtain the sorted predicted confidence set $\theta_c = [\theta_{c,1}, \theta_{c,2}, ..., \theta_{c,l_c}]$ of each category c, the length of confidence set belonging to category c can be calculated as follows:

$$\mathbf{l}_c = \sum_{n=1}^{N_t} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{1}[\widetilde{Y}_{n,i,j}^t = c], \tag{10}$$

where $\widetilde{Y}_{n,i,j}^t \in \mathbb{L}^{H \times W}, \mathbb{L} = \{1, 2, \cdots, C\}$ is the predicted pseudo-label for the image x_n^t. Then, each class threshold of pseudo-labels can be obtained by fixed percentage of the ranked confidence sets, where the percentage is denoted as a hyper-parameter η.

In addition to above self-training strategy, there are some works [30, 48, 50] focusing on self-training itself improvements, like ProDA [48] which leverages prototypes to obtain accurate pseudo-label. Since our proposed ProCA mainly works during feature adaptation process, which is orthogonal to such self-training based improvements. Thus we could combine our ProCA with such self-training methods to achieve better performance, which is shown in Table 6.

4 Experiments

Datasets and Evaluation Metrics: Following previous works [21, 43], we evaluate the model in common UDA of semantic segmentation benchmarks, GTA5 [35] → Cityscapes [4] and SYNTHIA [36] → Cityscapes [4]. GTA5 is an image dataset synthesized by a photo-realistic open-world computer game.

Table 1. Comparison results of **GTA5 → Cityscapes**. All methods use DeepLab-v2 with ResNet-101 backbone for **fair comparison**. † means that we report the first stage self-training result of ProDA [48] for fair comparison, please see Table 3 of ProDA [48] for details.

Method	Venue	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	trunk	bus	train	motorbike	bike	mIoU	gain
Source Only	-	53.8	15.6	69.3	28.1	18.8	27.6	34.9	18.2	82.5	27.8	71.6	59.4	35.3	44.1	25.9	37.5	0.1	28.9	24.9	37.3	+0.0
PatchAlign [41]	CVPR'19	92.3	51.9	82.1	29.2	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5	+9.2
ADVENT [42]	CVPR'19	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	+8.2
BDL [22]	CVPR'19	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	+11.2
UIDA [32]	CVPR'20	90.6	37.1	82.6	30.1	19.1	29.5	32.4	20.6	85.7	40.5	79.7	58.7	31.1	86.3	31.5	48.3	0.0	30.2	35.8	46.3	+9.0
LTIR [19]	CVPR'20	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2	+12.9
PIT [28]	CVPR'20	87.5	43.4	78.8	31.2	30.2	36.3	39.9	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	+13.2
LSE [38]	ECCV'20	90.2	40.0	83.5	31.9	26.4	32.6	38.7	37.5	81.0	34.2	84.6	61.6	33.4	82.5	32.8	45.9	6.7	29.1	30.6	47.5	+10.2
WeakSeg [34]	ECCV'20	91.6	47.4	84.0	30.4	28.3	31.4	37.4	35.4	83.9	38.3	83.9	61.2	28.2	83.7	28.8	41.3	8.8	24.7	46.4	48.2	+10.9
CrCDA [15]	ECCV'20	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6	+11.3
FADA [43]	ECCV'20	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	+11.9
IAST [30]	ECCV'20	**94.1**	**58.8**	85.4	39.7	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	**50.3**	24.7	35.2	40.2	52.2	+14.9
ASA [54]	TIP'21	89.2	27.8	81.3	25.3	22.7	28.7	36.5	19.6	83.8	31.4	77.1	59.2	29.8	84.3	33.2	45.6	16.9	34.5	30.8	45.1	+7.8
CLAN [26]	TPAMI'21	88.7	35.5	80.3	27.5	25.0	29.3	36.4	28.1	84.5	37.0	76.6	58.4	29.7	81.2	38.8	40.9	5.6	32.9	28.8	45.5	+8.2
DACS [39]	WACV'21	89.9	39.7	**87.9**	39.7	39.5	38.5	46.4	52.8	**88.0**	**44.0**	**88.8**	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1	+14.8
RPLL [50]	IJCV'21	90.4	31.2	85.1	36.9	25.6	37.5	**48.8**	48.5	85.3	34.8	81.1	64.4	36.8	86.3	34.9	52.2	1.7	29.0	44.6	50.3	+13.0
DAST [47]	AAAI'21	92.2	49.0	84.3	36.5	28.9	33.9	38.8	28.4	84.9	41.6	83.2	60.0	28.7	87.2	45.0	45.3	7.4	33.8	32.8	49.6	+12.3
ConTrans [20]	AAAI'21	95.3	65.1	84.6	33.2	23.7	32.8	32.7	36.9	86.0	41.0	85.6	56.1	25.9	86.3	34.5	39.1	11.5	28.3	43.0	49.6	+13.2
CIRN [7]	AAAI'21	91.5	48.7	85.2	33.1	26.0	32.3	33.8	34.6	85.1	43.6	86.9	62.2	28.5	84.6	37.9	47.6	0.0	35.0	36.0	49.1	+11.8
SDCA [21]	Arxiv'21	92.8	52.5	85.9	34.8	28.1	40.3	44.4	33.4	86.7	41.7	87.1	67.4	37.3	88.1	39.9	52.5	1.4	34.2	55.0	52.9	+15.6
PWCL [23]	Arxiv'21	93.3	54.2	83.0	25.9	28.1	37.2	41.1	39.3	83.1	38.9	78.2	61.3	36.2	84.2	35.8	54.0	18.1	26.7	47.5	50.9	+13.6
CLST [29]	Arxiv'21	92.8	53.5	86.1	39.1	28.1	28.9	43.6	39.4	84.6	35.7	88.1	63.9	38.3	86.0	41.6	50.6	0.1	30.4	51.7	51.6	+14.3
ESL [37]	CVPR'21	90.2	43.9	84.7	35.9	28.5	31.2	37.9	34.0	84.5	42.2	83.9	59.0	32.2	81.8	36.7	49.4	1.8	30.6	34.1	48.6	+11.3
MetaCorrect [9]	CVPR'21	92.8	58.1	86.2	39.7	33.1	36.3	42.0	38.6	85.5	37.8	87.6	62.8	31.7	84.8	35.7	50.3	2.0	36.8	48.0	52.1	+14.8
ProDA† [48]	CVPR'21	91.5	52.3	82.9	**42.0**	35.7	40.0	44.4	**43.2**	87.0	43.8	79.5	66.4	31.3	86.7	41.1	52.5	0.0	**45.4**	53.8	53.7	+16.4
UPLR [45]	ICCV'21	90.5	38.7	86.5	41.1	32.9	40.5	48.2	42.1	86.5	36.8	84.2	64.5	38.1	87.2	34.8	50.4	0.2	41.8	54.6	52.6	+15.3
Ours	-	91.9	48.4	87.3	41.5	31.8	**41.9**	47.9	36.7	86.5	42.3	84.7	**68.4**	**43.1**	**88.1**	39.6	48.8	**40.6**	43.6	**56.9**	**56.3**	**+19.0**

which shares 19 classes with Cityscapes. It has 24,966 images with the resolution 1914 × 1052. SYNTHIA is a synthetic urban scene dataset. Following previous works [40], we use the subset SYNTHIA-RAND-CITYSCAPES sharing 16 common classes with Cityscapes. It contains 9400 images with the resolution 1280 × 760. Cityscapes is a dataset of real urban scenes, which is collected from 50 cities in Germany and neighboring cities. It has 2,975 training images, 500 validation images, and 1,525 test images, with the resolution 2048 × 1024. We report the results on Cityscapes validation set using the category-wise Intersection over Union (IoU). Specifically, we report the mean IoU (mIoU) of all 19 classes in GTA5 → Cityscapes setting and the 16 common categories in SYNTHIA → Cityscapes setting. In addition, since some works [26, 40] only report mIoU for 13 common categories in SYNTHIA → CItyscapes setting, we also report the 13 common categories performance denoted as mIoU*.

Implementation Details. Following most previous works [12, 21, 43], we use the DeepLab-v2 framework [2] with ResNet-101 [11] encoder as our segmentation model for fair comparison. All models are pre-trained on ImageNet [5]. Atrous Spatial Pyramid Pooling (ASPP) [2] is inserted after the last encoder layer with dilated rates {6, 12, 18, 24}. At last, an up-sampling layer is used to obtain the final per-pixel predictions with the same image size as input. We implement the proposed method with PyTorch [33] on NVIDIA Tesla V100. We apply SGD optimizer with the initial learning rate of 2.5×10^{-4}, momentum 0.9 and weight

Table 2. Ablation studies of each component for GTA5 → Cityscapes. F refers to feature-level prototypical contrast adaptation; O refers to output-level prototypical contrast adaptation; Ada-ST refers to adaptive threshold self-training; MST refers to multi-scale testing. All methods use DeepLab-v2 with ResNet-101 backbone.

Source Only	F	O	Ada-ST	MST	mIoU
✓					37.3
✓	✓				47.9
✓		✓			48.4
✓	✓	✓			48.8
✓			✓		43.9
✓	✓	✓	✓		55.1
✓	✓	✓	✓	✓	**56.3**

Table 3. Ablation studies of different domain alignment methods for GTA5 → Cityscapes. FADA [43] refers fine-grained adversarial training for feature-level and output-level; SDCA [21] refers semantic distribution-aware adaptation; Memory Bank refers to pixel-level bank for contrast adaptation. ProCA refers to prototypical contrast adaptation. All methods use DeepLab-v2 with ResNet-101 backbone.

Source Only	FADA	SDCA	Memory Bank	ProCA	mIoU
✓					37.3
✓	✓				46.9
✓		✓			47.2
✓			✓		47.6
✓				✓	**48.8**

decay of 5.0×10^{-4}. We use polynomial learning rate scheduling with the power of 0.9. During prototypical contrast adaptation, the pseudo-label threshold of target domain is set to 0.9. For self-training stage. we assign pseudo-labels based on the predicted category probabilities with the adaptive thresholds. The percentage η of the number of pixels for each category is 0.6 in default.

4.1 Comparisons with State-of-the-Art Methods

In order to compare with previous state-of-the-art methods comprehensively, we include two typical methods: 1) Domain alignment methods which aim to align the distribution between source and target domains by distribution distances or adversarial training, including LITR [19], PIT [28], WeakSeg [34], CrCDA [15], FADA [43], ASA [54], CLAN [26], ConTrans [20], SDCA [21], and CIRN [7]. 2) Self-training approaches, including UIDA [32], LSE [38], IAST [30], DACS [39], RPLL [50], DAST [47], ESL [37], MetaCorrect [9], and ProDA [48].

Results on GTA5 → Cityscapes. As shown in Table 1, our approach achieves 56.3 % mIoU, outperforming prior methods by a large margin. In particular, the most challenging classes stated in [21] including pole, person, rider, bike, and train, obtains the significant improvements, compared to previous work. It

Table 4. Comparison results of **SYNTHIA → Cityscapes**. mIoU* denotes the mean IoU of 13 classes, which excludes the classes marked by the asterisk. All methods use DeepLab-v2 with ResNet-101 backbone for fair comparison. † means that we report the first stage self-training result of ProDA [48] for fair comparison. The result of ProDA [48] is from their released code.

Method	Venue	road	sidewalk	building	wall*	fence*	pole*	light	sign	vegetation	sky	person	rider	car	bus	motorbike	bike	mIoU	gain	mIoU*	gain*
Source Only	-	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	0.0	38.6	0.0
PatchAlign [41]	CVPR'19	82.4	38.0	78.6	-	-	-	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	-	-	46.5	+7.9
ADVENT [42]	CVPR'19	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	+7.7	48.0	+9.4
BDL [22]	CVPR'19	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	+12.8
UIDA [32]	CVPR'20	84.3	37.7	79.5	5.3	0.4	24.9	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	41.7	+8.2	48.9	+10.3
LTIR [19]	CVPR'20	92.6	53.2	79.2	-	-	-	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	-	-	49.3	+10.7
PIT [28]	CVPR'20	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	44.0	+10.5	51.8	+13.2
LSE [38]	ECCV'20	82.9	43.1	78.1	9.3	0.6	28.2	9.1	14.4	77.0	83.5	58.1	25.9	71.9	38.0	29.4	31.2	42.6	+9.1	49.4	+10.8
CrCDA [15]	ECCV'20	86.2	44.9	79.5	8.3	0.7	27.8	9.4	11.8	78.6	86.5	57.2	26.1	76.8	39.9	21.5	32.1	42.9	+9.4	50.0	+11.4
WeakSeg [34]	ECCV'20	92.0	53.5	80.9	11.4	0.4	21.8	3.8	6.0	81.6	84.4	60.8	24.4	80.5	39.0	26.0	41.7	44.3	+10.8	51.9	+13.3
IAST [30]	ECCV'20	81.9	41.5	83.3	17.7	**4.6**	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	**33.1**	52.7	49.8	+16.3	57.0	+18.4
FADA [43]	ECCV'20	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	45.2	+11.7	52.5	+13.9
ASA [54]	TIP'21	91.2	48.5	80.4	3.7	0.3	21.7	5.5	5.2	79.5	83.6	56.4	21.9	80.3	36.2	20.0	32.9	41.7	+8.2	49.3	+10.7
CLAN [26]	TPAMI'21	82.7	37.2	81.5	-	-	-	17.1	13.1	81.2	83.3	55.5	22.1	76.6	30.1	23.5	30.7	-	-	48.8	+10.2
DACS [39]	WACV'21	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	**90.8**	67.6	**38.3**	82.9	38.9	28.5	47.6	48.3	+14.8	54.8	+16.2
RPLL [50]	IJCV'21	87.6	41.9	83.1	14.7	1.7	36.2	31.3	19.9	81.6	80.6	63.0	21.8	86.2	40.7	23.6	53.1	47.9	+14.4	54.9	+16.3
CIRN [7]	AAAI'21	85.8	40.4	80.4	4.7	1.8	30.8	16.4	18.6	80.7	80.4	55.2	26.3	83.9	43.8	18.6	34.3	43.9	+10.4	51.1	+12.5
DAST [47]	AAAI'21	87.1	44.5	82.3	10.7	0.8	29.9	13.9	13.1	81.6	86.0	60.3	25.1	83.1	40.1	24.4	40.5	45.2	+11.7	52.5	+13.9
ConTrans [20]	AAAI'21	**93.3**	**54.0**	81.3	14.3	0.7	28.8	21.3	22.8	82.6	83.3	57.7	22.8	83.4	30.7	20.2	47.2	46.5	+13.0	53.9	+15.3
SDCA [21]	Arxiv'21	88.4	45.9	83.9	24.0	1.7	38.1	25.2	17.0	85.3	82.9	67.3	26.6	87.1	47.2	28.6	53.4	50.2	+16.7	56.8	+18.2
PWCL [23]	Arxiv'21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	53.3	+14.7
CLST [29]	Arxiv'21	88.0	49.2	82.2	16.3	0.4	29.2	31.8	23.9	84.1	88.0	59.1	27.2	85.5	46.4	28.9	**56.5**	49.8	+16.3	57.8	+19.2
ESL [37]	CVPR'21	84.3	39.7	79.0	9.4	0.7	27.7	16.0	14.3	78.3	83.8	59.1	26.6	72.7	35.8	23.6	45.8	43.5	+10.0	50.7	+12.1
MetaCorrect [9]	CVPR'21	92.6	52.7	81.3	8.9	2.4	28.1	13.0	7.3	83.5	85.0	60.1	19.7	84.8	37.2	21.5	43.9	45.1	+11.6	52.5	+13.9
ProDA† [48]	CVPR'21	87.1	44.0	83.2	26.9	0.0	**42.0**	**45.8**	**34.2**	86.7	81.3	68.4	22.1	87.7	50.0	31.4	38.6	51.9	+18.4	58.5	+19.9
UPLR [45]	ICCV'21	79.4	34.6	83.5	19.3	2.8	35.3	32.1	26.9	78.8	79.6	66.6	30.3	86.1	36.6	19.5	56.9	48.0	+14.5	54.6	+16.0
Ours	-	90.5	52.1	**84.6**	**29.2**	3.3	40.3	37.4	27.3	**86.4**	85.9	**69.8**	28.7	**88.7**	**53.7**	14.8	54.8	**53.0**	**+19.5**	**59.6**	**+21.0**

Table 5. Ablation studies of different prototypes updating scheme for GTA5 → Cityscapes. Fixed refers to no-updating for calculated prototypes; Source means updating in a strict statistical way on in source domain as Equation 7; Mixed refers updating in Equation 8 in both source and target domain.

Source Only	Fixed	Source	Mixed	mIoU
✓				37.3
✓	✓			47.8
✓		✓		48.3
✓			✓	**48.8**

demonstrates our motivation that the inter-class modeling via prototypes indeed help the category recognition on the target domain, especially for the harder classes.

Results on SYNTHIA → Cityscapes. The comparisons of SYNTHIA → Cityscapes are shown in Table 4. Among all the 16 categories, we achieve the best scores on 6 categories, most of those are hard classes stated in [21], *e.g.*, person, and bike. To be specific, the proposed method achieves the mIoU score by 53.0% and 59.6% over the 16 and 13 categories respectively, which obtains the gains over the baseline by 19.5% and 21.0%.

Table 6. Ablation studies of different self-training schemes for GTA5 → Cityscapes. Naive Self-Training refers to fixed 0.9 threshold for pseudo-label generation; Adaptive Self-Training refers to adaptive pseudo-label generation, which is median of predicted confidence set of each class in default (Sec 3.3). Prototypes-based Self-Training refers to pseudo-label generation strategy by utilizing prototypes which is proposed by ProDA [48].

ProCA	Naive	Adaptive	Prototypes-based	mIoU
✓				48.8
✓	✓			55.2
✓		✓		56.3
✓			✓	57.5

Discussion with ProDA. It should be noticeable that our proposed prototype contrastive learning method surpass a similar prototype-based method ProDA [48] on both transferring scenarios under a *fair* comparison setting. Especially, in GTA5 → Cityscapes, our adaptive method outperforms [48] by a large margin of 1.4% mIOU. This is due to the fact that ProDA only utilizes prototypes to rectify pseudo-labels or align feature in a purely sample-wise manner, which is more vulnerable to the interference from outlier or noisy samples in the target domain, while our pipeline directly depicts the class-wise relation in a sample-to-prototype manner, making the learning process more robust and friendly to cross-domain transferring.

Discussion with Other Contrastive Learning based Methods. It should also be noticed that compared with a similar patch-wise contrastive learning method PWCL [23], our approach achieves superiority of 4.2% and 5.4% mIOU improvement on both GTA5 → Cityscapes and SYNTHIA → Cityscapes respectively. This is due to the fact that PWCL only takes patch-wise features for contrastive feature adaptation, which is coarse to depict class-wise relation and ignores the fine-grained pixel-wise distribution variation during training process, resulting in less discriminative and general representation.

How ProCA helps poor classes adaptation? As shown in Table1, the performance of *train* class could not be improved by state-of-the-art pseudo-label method ProDA. This is because initialized predictions are totally wrong, thus ProDA could not estimate accurate pseudo-label for *train* class. Different from ProDA, our ProCA first corrects the *train* class predictions by push aware from others class centroids, which progressively obtain more and more accurate feature representation of *train* class. After introducing such relationship between different classes, our proposed method achieves highest *train* class performance after combining with self-training method.

4.2 Ablation Studies

Effectiveness of Each Component. We conduct ablation studies to demonstrate the effectiveness of each component. We use the ResNet-101 backbone

Table 7. Ablation studies of different percentages determining class-wise thresholds during self-training process for GTA5 → Cityscapes. All methods use DeepLab-v2 with ResNet-101 backbone.

η (%)	30	40	50	60	70	80	90
mIoU	54.3	54.7	54.9	55.1	54.4	54.1	52.5

Table 8. Ablation studies of different contrastive adaptation choices for GTA5 → Cityscapes. $s \rightarrow s$ means Eq. 5 and $t \rightarrow s$ means Eq. 4

Source Only	$s \rightarrow s$	$t \rightarrow s$	mIoU
✓			37.3
✓	✓		44.9
✓		✓	46.8
✓	✓	✓	48.8

with DeepLab-v2 segmentation for GTA5 → Cityscapes adaptation. As shown in Table 2, the source-only baseline achieves 37.3% mIoU on Cityscapes val set. Further, we achieve 48.8% mIoU score after using the proposed prototypical contrast adaptation. At last, the performance can be improved to 55.1% mIoU through self-training with class-aware adaptive thresholds. Finally, we obtain 56.3% mIoU score by multi-scale testing following FADA [43]. When directly using self-training after source-domain training, we could only obtain 43.9% mIoU, which is 11.2% mIoU lower than 55.1% mIoU score, demonstrating the effectiveness of ProCA.

Effectiveness of ProCA. To verify the effectiveness of ProCA, we implement other feature alignment methods, *e.g*, class-wise adversarial training without inter-class modeling FADA [43], semantic-distribution modeling with category-wise information. As shown in Table 3, FADA improves the baseline to 46.9% mIoU, which indicates the effectiveness of the adversarial training. SDCA [21] obtains 47.2% mIoU by considering semantic-aware feature alignment. Memory Bank obtains 47.6% mIoU by introducing pixel-wise contrastive adaptation, which already achieves better performance than FADA and SDCA. Compared with above methods, our ProCA achieves the best mIoU score 48.8%, which demonstrates the superiority of the proposed class-aware prototypical contrast adaptation than pixel-wise memory bank scheme.

Effectiveness of Mixed Updating. We conduce ablation studies to verify the effectiveness of mixed updating for prototypes. As shown in 5, a naive fixed prototype scheme only achieves 47.8% mIoU, while centroid updating way only in source domain obtains 48.3% mIoU, which has 0.5% gain compared with fixed-prototype scheme. Mixed updating scheme achieves best 48.8% mIoU score, which demonstrates the effectiveness of latest features during training.

Effectiveness of Multi-Level Adaptation. We conduct ablation studies to verify the effectiveness of multi-level adaptation. The results are shown in Table 2. when only using feature-level adaptation or output-level adaptation, we

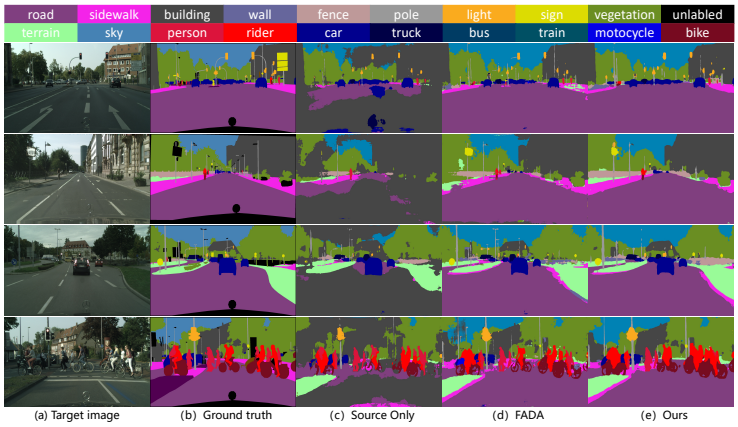

Fig. 3. Qualitative segmentation results for GTA5 → Cityscapes. From the left to right: target image, ground-truth, predictions by Source Only, FADA [43] and our proposed method are shown.

achieve 47.9% mIoU and 48.4% mIoU, respectively. After combining them, we obtain the best mIoU score 48.8%, demonstrating the superiority of multi-level adaptation.

Effectiveness of In-Domain Contrastive Adaptation and Cross-Domain Contrastive Adaptation. We conduct experiments to study the influence of different domain choices of prototypical contrastive adaptation. The results are shown in Table 8. When only using source-to-source ProCA scheme, we could obtain 7.6% mIoU improvement. When only using cross-domain ProCA scheme, we could obtain 9.5% mIoU improvement. After combining both in-domain and cross-domain strategies, we finally obtain 48.8% mIoU, which verifies the effectiveness of the proposed method.

Effectiveness of Different Percentages for Adaptive Self-training. We conduct experiments to study the influence of different percentages of pseudo-labels generation during self-training stage. The results are shown in Table 7. Using 60 percentage to generate pseudo-labels, ProCA achieves the best mIoU 55.1%. And larger percentages harm the performance.

5 Conclusions

In this paper, we propose ProCA, which utilizes class-wise prototypes to align features in a fine-grained manner. Apart from feature-level adaptation, output-level prototypes are also exploited to boost the adaptation performance. The proposed method achieves the state-of-the-art performance on challenging benchmarks, outperforming previous methods by a large margin. Elaborate ablative studies demonstrate the advancement of our ProCA. We hope the proposed prototypical contrast adaptation could extend to more tasks, such as object detection and instance segmentation.

References

1. Araslanov, N., Roth, S.: Self-supervised augmentation consistency for adapting semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 15384–15394 (2021)
2. 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 Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2017)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)
4. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)
7. Gao, L., Zhang, L., Zhang, Q.: Addressing domain gap via content invariant representation for semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 7528–7536 (2021)
8. Gu, Q., Zhou, Q., Xu, M., Feng, Z., Cheng, G., Lu, X., Shi, J., Ma, L.: Pit: Position-invariant transform for cross-fov domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8761–8770 (2021)
9. Guo, X., Yang, C., Li, B., Yuan, Y.: Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3927–3936 (2021)
10. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
12. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning. pp. 1989–1998. PMLR (2018)
13. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
14. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017)
15. Huang, J., Lu, S., Guan, D., Zhang, X.: Contextual-relation consistent domain adaptation for semantic segmentation. In: European Conference on Computer Vision. pp. 705–722 (2020)

16. Jiang, Z., Gao, P., Guo, C., Zhang, Q., Xiang, S., Pan, C.: Video object detection with locally-weighted deformable neighbors. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
17. Jiang, Z., Gu, Z., Peng, J., Zhou, H., Liu, L., Wang, Y., Tai, Y., Wang, C., Zhang, L.: Stc: Spatio-temporal contrastive learning for video instance segmentation. arXiv preprint arXiv:2202.03747 (2022)
18. Jiang, Z., Liu, Y., Yang, C., Liu, J., Gao, P., Zhang, Q., Xiang, S., Pan, C.: Learning where to focus for efficient video object detection. In: European conference on computer vision. pp. 18–34. Springer (2020)
19. Kim, M., Byun, H.: Learning texture invariant representation for domain adaptation of semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12975–12984 (2020)
20. Lee, S., Hyun, J., Seong, H., Kim, E.: Unsupervised domain adaptation for semantic segmentation by content transfer. arXiv preprint arXiv:2012.12545 (2020)
21. Li, S., Xie, B., Zang, B., Liu, C.H., Cheng, X., Yang, R., Wang, G.: Semantic distribution-aware contrastive adaptation for semantic segmentation. arXiv preprint arXiv:2105.05013 (2021)
22. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6936–6945 (2019)
23. Liu, W., Ferstl, D., Schulter, S., Zebedin, L., Fua, P., Leistner, C.: Domain adaptation for semantic segmentation via patch-wise contrastive learning. arXiv preprint arXiv:2104.11056 (2021)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)
25. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning. pp. 97–105. PMLR (2015)
26. Luo, Y., Liu, P., Zheng, L., Guan, T., Yu, J., Yang, Y.: Category-level adversarial adaptation for semantic segmentation using purified features. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
27. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2507–2516 (2019)
28. Lv, F., Liang, T., Chen, X., Lin, G.: Cross-domain semantic segmentation via domain-invariant interactive relation transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4334–4343 (2020)
29. Marsden, R.A., Bartler, A., Döbler, M., Yang, B.: Contrastive learning and self-training for unsupervised domain adaptation in semantic segmentation. arXiv preprint arXiv:2105.02001 (2021)
30. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: European Conference on Computer Vision. pp. 415–430 (2020)
31. Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., Kim, K.: Image to image translation for domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4500–4509 (2018)
32. Pan, F., Shin, I., Rameau, F., Lee, S., Kweon, I.S.: Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3764–3773 (2020)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems pp. 8026–8037 (2019)
34. Paul, S., Tsai, Y.H., Schulter, S., Roy-Chowdhury, A.K., Chandraker, M.: Domain adaptive semantic segmentation using weak labels. In: European Conference on Computer Vision. pp. 571–587 (2020)
35. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European Conference on Computer Vision. pp. 102–118 (2016)
36. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3234–3243 (2016)
37. Saporta, A., Vu, T.H., Cord, M., Pérez, P.: Esl: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation. arXiv preprint arXiv:2006.08658 (2020)
38. Subhani, M.N., Ali, M.: Learning from scale-invariant examples for domain adaptation in semantic segmentation. In: European Conference on Computer Vision. pp. 290–306 (2020)
39. Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: Dacs: Domain adaptation via cross-domain mixed sampling. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision. pp. 1379–1389 (2021)
40. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7472–7481 (2018)
41. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1456–1465 (2019)
42. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2517–2526 (2019)
43. Wang, H., Shen, T., Zhang, W., Duan, L.Y., Mei, T.: Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In: European Conference on Computer Vision. pp. 642–659 (2020)
44. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
45. Wang, Y., Peng, J., Zhang, Z.: Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9092–9101 (2021)
46. Xu, Q., Yao, L., Jiang, Z., Jiang, G., Chu, W., Han, W., Zhang, W., Wang, C., Tai, Y.: Dirl: Domain-invariant representation learning for generalizable semantic segmentation (2022)
47. Yu, F., Zhang, M., Dong, H., Hu, S., Dong, B., Zhang, L.: Dast: Unsupervised domain adaptation in semantic segmentation based on discriminator attention and

self-training. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 10754–10762 (2021)
48. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12414–12424 (2021)
49. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2881–2890 (2017)
50. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. International Journal of Computer Vision **129**(4), 1106–1120 (2021)
51. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 633–641 (2017)
52. Zhou, Q., Gu, Q., Pang, J., Feng, Z., Cheng, G., Lu, X., Shi, J., Ma, L.: Self-adversarial disentangling for specific domain adaptation. arXiv preprint arXiv:2108.03553 (2021)
53. Zhou, Q., Zhuang, C., Lu, X., Ma, L.: Domain adaptive semantic segmentation with regional contrastive consistency regularization. arXiv preprint arXiv:2110.05170 (2021)
54. Zhou, W., Wang, Y., Chu, J., Yang, J., Bai, X., Xu, Y.: Affinity space adaptation for semantic segmentation across domains. IEEE Transactions on Image Processing **30**, 2549–2561 (2020)
55. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232 (2017)
56. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: European Conference on Computer Vision. pp. 289–305 (2018)