Efficient Active Domain Adaptation for Semantic Segmentation by Selecting Information-rich Superpixels

Yuan Gao[1], Zilei Wang[1]✉, Yixin Zhang[1], and Bohai Tu[1]

MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China

Abstract. Unsupervised Domain Adaptation (UDA) for semantic segmentation has been widely studied to exploit the label-rich source data to assist the segmentation of unlabeled samples on target domain. Despite these efforts, UDA performance remains far below that of fully-supervised model owing to the lack of target annotations. To this end, we propose an efficient superpixel-level active learning method for domain adaptive semantic segmentation to maximize segmentation performance by automatically querying a small number of superpixels for labeling. To conserve annotation resources, we propose a novel low-uncertainty superpixel fusion module which amalgamates superpixels possessing low-uncertainty features based on feature affinity and thereby ensuring high-quality fusion of superpixels. As for the acquisition strategy, our method takes into account two types of information-rich superpixels: large-size superpixels with substantial information content, and superpixels with the greatest value for domain adaptation learning. Further, we employ the cross-domain mixing and pseudo label with consistency regularization techniques respectively to address the domain shift and label noise problems. Extensive experimentation demonstrates that our proposed superpixel-level method utilizes a limited budget more efficiently than previous pixel-level techniques and surpasses state-of-the-art methods at 40x lower cost. Our code is available at `https://github.com/EdenHazardan/ADA_superpixel`.

Keywords: Domain adaptation · Active learning · Superpixel-levels

1 Introduction

As a fundamental task in computer vision, semantic segmentation has made remarkable strides, enabling a multitude of applications such as autonomous driving [32], robotics [27], and disease diagnosis [34]. The significant advancements can be primarily attributed to the availability of extensively labeled datasets [10, 12, 22]. However, the process of labeling pixel-level segmentation data is laborious and expensive [13, 40], which poses a significant obstacle to practical advancements. To tackle this issue, researchers have proposed unsupervised domain adaptation (UDA) approaches [20, 46], which aim to adapt a

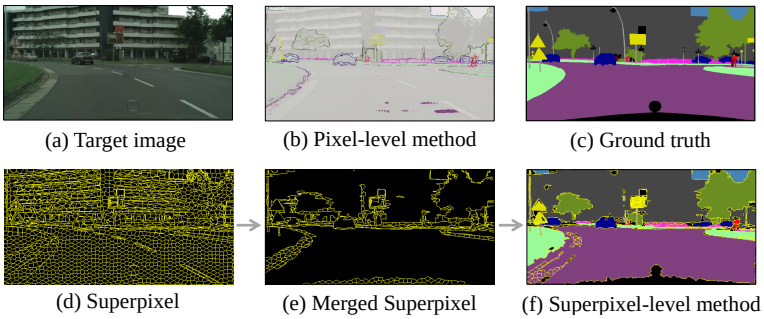

Fig. 1: Illustration of different active domain adaptation strategies. (b) Pixel-level methods (e.g., LabOR [31], RIPU [40]) use pixels as unit and select 11,535 (2.2%) or 26,215 (5%) pixels per image for labeling based on uncertainty. Instead, our method (d,e,f) chooses superpixels as unit and perform superpixel fusion. Impressively, we have achieved state-of-the-art performance by only using 640 labeling clicks per image.

model trained on a source domain with rich annotations to an unlabeled target domain. Although UDA methods have achieved impressive outcomes, it is worth noting that in the absence of target annotations, their performance is still far below that of full supervision on target domain [31,39,40]. Motivated by the limitation of UDA, we expect to enhance the model performance on target domain by labeling small amounts of target samples, and active learning technologies have the potential to accomplish precisely that.

Active learning aims to maximize the model performance with few informative labeled data. Indeed, active learning (AL) techniques have found widespread application in traditional semantic segmentation tasks. Many methods [4, 9, 24] employ a patch-based approach, dividing the image into non-overlapping patches and treating each patch as a sample. By utilizing a thoughtfully designed acquisition function, these methods select the most informative patches and proceed to pixel-level labeling. Recently, some work [3, 17] points out that pixel-level labeling is inefficient and costly. Instead, they propose superpixel-level methods, employing superpixels as the fundamental unit. During the labeling process, only one dominant label, obtained through a single click, is requested for each selected superpixel (containing many pixels), thereby greatly reducing the labeling cost. Specifically, RSAL [3] utilizes the SEEDS [2] algorithm to generate superpixels. ASAL [17] addresses the issue of over-segmentation in RSAL by proposing a superpixel merging technique based on feature affinity where all superpixels are involved in fusion process. With the rapid development of AL technology in semantic segmentation, recent works [31, 39, 40, 44] have integrated AL into domain adaptive semantic segmentation, resulting in active domain adaptation (ADA) task. However, these approaches primarily rely on pixel-level methods, treating each pixel as an individual sample, as shown in Fig. 1(b). By employing an acquisition function driven by uncertainty, they typically label a substantial

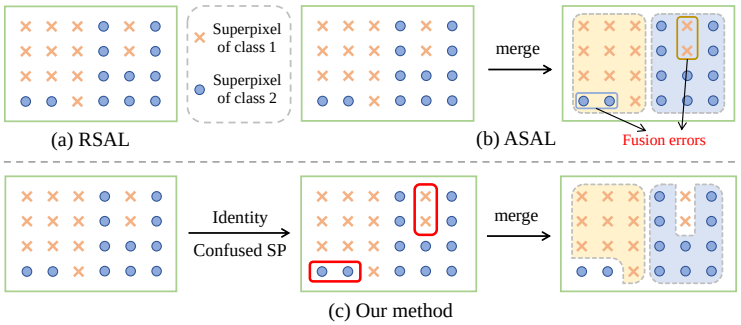

Fig. 2: Different superpixel-level methods. (a) RSAL [3] uses original superpixels which is over-segmented. (b) ASAL [17] proposes superpixel merging based on feature affinity, but fusion errors occur. (c) Our method identifies easily confused superpixels and excludes them during the fusion process, resulting in improved superpixel fusion quality.

number of pixels , such as $11,535\,(2.2\%)$ or $26,215\,(5\%)$ pixels, for each image with a resolution of 1024×512 to achieve good results. Such labeling costs are not feasible in practical scenarios. Therefore, our work aims to introduce superpixel-level AL methods to domain adaptive semantic segmentation.

An intuitive idea is to directly apply the superpixel-level ASAL [17] method to ADA task. The key steps involve merging superpixels, selecting and annotating them based on a carefully designed acquisition function, and subsequently training the final model. However, this direct application faces challenges at each step, including errors in superpixel fusion, inadequate acquisition functions, and issues related to domain shift and label noise during model training. First, in terms of superpixel mergence, ASAL method may occur superpixel fusion errors due to the inter-class feature confusion issue inherent in domain adaptation settings, leading to a decline in the overall quality of the annotations, shown in Fig. 2(b). As outlined in [7,14,19,40], domain adaptation suffers from inter-class confusion (e.g., road and sidewalk, bus and train) due to the domain shift problem [38]. Therefore, our objective is to create a benchmark that can accurately identify features that are prone to confusion. This enables us to exclude these particular superpixels during the fusion process. Only the superpixels with distinguishable features will undergo merging, effectively addressing the problem of superpixel fusion errors, as illustrated in Fig. 2(c). Second, regarding the acquisition function, ASAL fails to account for the issue of domain shift, whereas existing ADA methods [31,40,44] predominantly emphasize the pixel level rather than the superpixel level. In contrast, our approach aims to comprehensively consider two types of information-rich superpixels: large-size superpixels with substantial information content and valuable superpixels exhibiting substantial domain differences. Third, since superpixels are not completely accurate, superpixel-level annotations are inevitably noisy. Besides, the supervision on source data is af-

fected by domain shift issues. Therefore, our work introduces label denoise and domain adaptation techniques on model training.

In this work, we design a superpixel-level AL method for domain adaptive semantic segmentation. We propose a novel low-uncertainty superpixel fusion module where the superpixels are divided according to the uncertainty of the features extracted by the UDA model [14, 35]. Superpixels with low uncertainty (*i.e.*, not easily confused) features are selected for fusion, while those with high uncertainty features (*i.e.*, easily confused) are excluded from the fusion process and retain their original state. In this way, our work can effectively reduce the cost of superpixel-level labeling while ensuring the labeling quality. To strategically select the most valuable superpixel-level annotations for domain adaptation, we devised two distinct acquisition functions respectively considering superpixel sizes and domain differences. As for the model training, our work proposes to use cross-domain mixing [14, 35] and pseudo label [8, 25] with consistency regularization techniques [26] respectively to address the domain shift problem of source domain data and the label noise problem caused by superpixels. Remarkably, our approach achieves superior performance by utilizing only 640 clicks per image, surpassing the previous methods [39, 40] that relied on a significantly larger set of 26, 215 labeled pixels (*i.e.*, 26, 215 clicks) per image.

We summarize the contributions of this work as follows:

- We propose an efficient superpixel-level active learning method for domain adaptive semantic segmentation and propose a novel low-uncertainty superpixel fusion module specifically designed to mitigate the issue of superpixel fusion errors.
- We have developed two efficient acquisition functions for the comprehensive selection of information-rich superpixels. Additionally, we employ cross-domain mixing and pseudo label with consistency regularization to address the challenges of domain shift and label noise.
- We experimentally evaluate the effectiveness and efficiency of our proposed method, and the results on two challenging benchmarks demonstrate the superiority of our method to previous state-of-the-art methods in terms of annotation cost and model performance.

2 Related Work

Domain adaptative semantic segmentation (DASS). DASS has been extensively studied to tackle the challenges of pixel-level dense annotation and domain shift [14,26,35,36]. Recently, DASS approaches have utilized a self-training technique to retrain the network with the pseudo labels generated from confident predictions on the target domain. To regularize the training with pseudo labels, consistency regularization [26] based on data augmentation [33] or cross-domain mixup [35] is commonly adopted. More recently, DAFormer [14] employs Transformer instead of CNN as architecture and has achieved state-of-the-art performance. Although DASS has shown great results, the lack of annotations

on target domain makes the DASS models [23, 35, 45] far inferior to the fully-supervised model.

Active Learning for semantic segmentation (ALSS). To mitigate the expenses associated with labeling in semantic segmentation, ALSS selectively collects labels among unlabeled samples and utilizes diverse predefined labeling units to achieve this objective. Many methods [4, 9, 24] employ a patch-based approach, dividing the image into non-overlapping patches and treating each patch as a sample. They perform pixel-level annotation on the selected patch, resulting in excessive annotation costs. RSAL [3] proposed to use superpixel-level approach and demonstrate its effectiveness over the patch-based approach. ASAL [17] pointed out that the superpixels used in RSAL are over-segmented and proposed superpixel merging to avoid annotation wastes. However, ASAL is not suitable for DASS. Superpixel fusion errors would occur due to the inter-class feature confusion [7, 14, 19, 40] in domain adaptation. Unlike ASAL, where all superpixels are involved in fusion process, our method introduces specific criteria for selecting suitable superpixels to participate in fusion. We exclude superpixels that can be easily confused, thereby improving the quality of fusion.

Active domain adaptation (ADA) for semantic segmentation. Active domain adaptation for semantic segmentation can strike a balance between annotation cost and model performance by selectively labeling a few yet valuable samples on target domain. MADA [28] proposes a multi-anchor strategy to actively select a subset of images and annotate the entire image, which is probably inefficient [40]. Recent works [31, 39, 40, 44] adopt pixel-level annotations. LabOR [31] selects pixels for labeling based on the difference between the prediction of two distinct classifiers. RIPU [31] constructs acquisition function in terms of uncertainty and region impurity. UBD [44] focuses on the boundary pixels. D2ADA [39] dynamically considers uncertainty and domain density. Despite good results, the labeling cost is too high and impractical, *i.e.*, $11,535\,(2.2\%)$ or $26,215\,(5\%)$ pixels, for each image with a resolution of 1024×512. Besides, the proposed acquisition functions are all designed for pixel unit.

In this work, we use superpixel-level annotations and propose acquisition functions suitable for superpixels. With labeling only 640 superpixels (*i.e.*, 640 clicks) per image, our method can outperform the existing methods [39, 40].

3 Our Approach

In this section, we first present an overview of our method. Then we detail each component of the framework, including superpixel generation (Section 3.2), low-uncertainty superpixel fusion (Section 3.3), acquisition function (Section 3.4) and training process (Section 3.5).

3.1 Overall Framework

Active domain adaptation for semantic segmentation aims to train a segmentation model that can perform well on target domain using a combination of

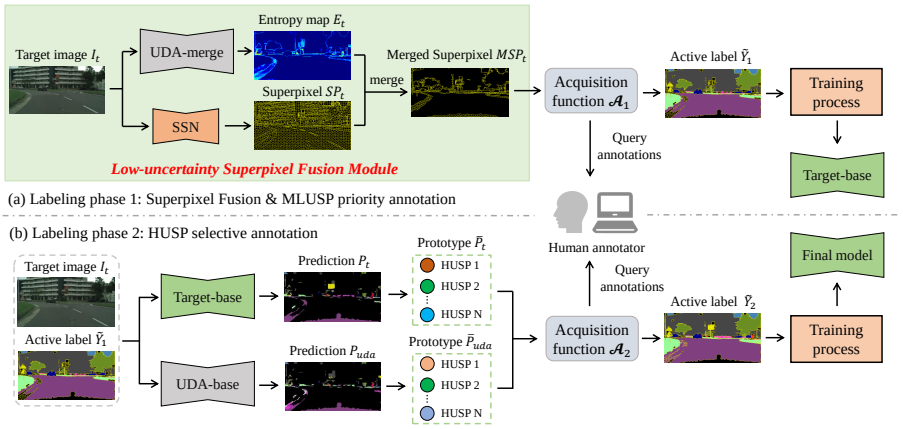

Fig. 3: The overview of our method. At labeling phase 1 (a), we adopt low-uncertainty superpixel fusion and select the merged low-uncertainty superpixels (MLUSPs) with large-size and query annotations from a human annotator. At labeling phase 2 (b), we select the high-uncertainty superpixels (HUSPs) with large domain differences based on the acquisition function $\mathcal{A}_2$ and query annotations. The final model is trained by our proposed training process with the selected active labels.

labeled samples from the source domain $S = \{I_s, Y_s\}$, partially labeled samples from target domain $T = \{I_t, \widetilde{Y}_t\}$. Here I refers to images, Y_s represents source annotations, and $\widetilde{Y}_t$ means the thoughtfully selected active labels on target domain. In this work, we first divide each image into superpixels. Next, we employ superpixel fusion and adopt acquisition functions to select a few informative superpixels which are then annotated by an oracle as $\widetilde{Y}_t$. Here, we use the ground truth segmentation label to simulate such an annotation process. Specifically, we use a dominant labeling scheme where each superpixel is assigned only a single class label [3] (*i.e.*, only one click). Finally, we train the model using S and T with cross-domain mixing and pseudo label with consistency regularization techniques.

3.2 Superpixel Generation

Superpixels are image primitives that group similar pixels and preserve object boundary well [17]. As a result, most pixels within a superpixel are from the same semantic category. This enables the use of a light-weight annotation scheme where each superpixel is annotated by only one class label that represents the majority of the pixels [3]. Superpixel generation algorithms can be broadly classified into traditional and CNN-based approaches. Traditional methods [1, 2, 21] utilize original features such as color and spatial position for clustering. In contrast, CNN-based methods [15,42], which use deep CNNs to learn features for superpixel generation, generally outperform traditional methods. However, CNN-

based approaches require ground truth semantic segmentation labels for each pixel to compute the learning loss [3]. RSAL [3] and ASAL [17] used the traditional algorithms SEEDS [2] to obtain superpixels. In our task, semantic segmentation labels are available on source domain, so we can use a CNN-based SSN [15] that has been pre-trained on the source domain to obtain better superpixel results on target domain.

3.3 Low-uncertainty Superpixel Fusion

The original superpixel results are over-segmented [17]. To save the annotation budget, ASAL [17] proposes to fuse the superpixels with similar predictions. However, in DASS task, fusion errors would occur due to the inter-class feature confusion issue. To tackle this issue, our method proposes to identify the superpixels with easily confused features and exclude them during the fusion process. Specifically, we utilize the classic uncertainty metric entropy to partition superpixels. As shown in Fig. 3(a), we use an off-the-shelf UDA model (UDA-merge) and SSN model respectively extract the entropy map E_t and superpixel SP_t on target domain. Then, a base superpixel $s \in SP_t$ is divided into two categories according to the entropy map: low uncertainty superpixel (LUSP) or high uncertainty superpixel (HUSP):

$$s = \begin{cases} \text{LUSP}, & ent_s \leq \tau \\ \text{HUSP}, & ent_s > \tau \end{cases} \tag{1}$$

where $ent_s = \frac{\sum_{x \in s} E_t(x)}{|\{x : x \in s\}|}$ is the averaged entropy of superpixel $s \in SP_t$, and τ is a threshold. Typically, the features of LUSP are more distinguishable, making LUSP more suitable for fusion. On the other hand, the features of HUSP are more easily confused, and thus HUSP should be excluded from fusion. Therefore, we perform superpixel fusion on LUSP. Following ASAL [17], we employ the square root of Jensen-Shannon (JS) divergence as a symmetric measure of discrepancy between two prediction distributions of superpixels. To be specific, any two LUSP s, n would be amalgamated together only if

$$d_{JS}(f(s)||f(n)) < \varepsilon, \tag{2}$$

where $f(s) = \frac{\sum_{x \in s} f(x)}{|\{x : x \in s\}|}$ is the averaged class prediction of LUSP s, $f(x)$ is the UDA-merge model's estimation of class probability on pixel x in superpixel s, and ε is a threshold. In this way, our proposed low-uncertainty superpixel fusion module outputs high-quality merged superpixels MSP_t, where LUSPs are fused and HUSPs remain unchanged.

3.4 Acquisition Function

In this work, we proposed two distinct acquisition functions $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively considering superpixel sizes and domain differences.

After superpixel fusion, MSP_t contains a small number of merged LUSPs (MLUSPs) with large-size and a large amount of HUSPs with original small size. Specifically, a target image I_t with a resolution of 1024×512 is divided into approximately $4,985$ superpixels, and each original superpixel contains about 105 pixels. According to the statistics, there are approximately $4,135$ low-uncertainty superpixels (LUSPs) and 850 high-uncertainty superpixels (HUSPs) per image. After fusion, the merged superpixels MSP_t contain about 23 MLUSPs with an average size of $18,699$ pixels and 850 HUSPs retaining their size of 105 pixels.

In segmentation tasks, both the quantity of pixels and uncertainty are important factors in selecting samples. However, as the above statistics show, the merged superpixels (MLUSPs) contain far more pixels than unmerged superpixels (HUSPs). Therefore, we first consider labeling information-rich MLUSPs with large-size. Then, as for the uniformly sized HUSPs, we utilize the domain differences as a benchmark to select the most valuable samples for domain adaptation learning. In this way, our method contains two labeling phases.

Acquisition function $\mathcal{A}_1$: As shown in Fig. 3(a), the acquisition function $\mathcal{A}_1$ chooses the information-rich superpixels with largest size, and thereby we label all MLUSPs and obtain $\widetilde{Y}_1$ at labeling phase 1. In fact, we only need 23 clicks (*i.e.*, labeling all the MLUSPs) for each image to get around 82.03% of the pixel annotations in the entire image.

Acquisition function $\mathcal{A}_2$: At labeling phase 2, we select the information-rich HUSPs with largest domain differences, which can be represented by the differences in predictions from two different domain models. Specifically, we use an off-the-shelf UDA model (UDA-base) as source domain model and utilize the model (Target-base) trained on $\widetilde{Y}_1$ as target domain model. For each HUSP, we use Target-base and UDA-base models to respectively extract the predictions and average the superpixel internal predictions to get the prototype $\overline{P}_t$ and $\overline{P}_{uda}$. For an HUSP s, we calculate its domain difference score ($DDscore_s$) according to the cosine similarity of its corresponding $\overline{P}_t$ and $\overline{P}_{uda}$:

$$DDscore_s = 1 - \frac{\overline{P}_t(s) \cdot \overline{P}_{uda}(s)}{\mid \overline{P}_t(s) \mid\mid \overline{P}_{uda}(s) \mid}. \tag{3}$$

The smaller the similarity of predictions from the two models, the greater the domain differences, and the higher the corresponding $DDscore$. Therefore, the acquisition function $\mathcal{A}_2$ selects the information-rich HUSPs with the highest $DDscore$ and obtains $\widetilde{Y}_2$ in labeling phase 2, as shown in Fig. 3(b). Finally, we train the final model using $\widetilde{Y}_2$.

3.5 Training Process

With fully-labeled samples on source domain and actively selected and annotated superpixels on target domain, we aim to train a network that can perform well on target domain. An intuitive method is to use all labeled data from source and target domain to train network by optimizing the standard supervised loss, like [40, 44]:

$$L_{sup} = L_{CE}(I_s, Y_s) + L_{CE}(I_t, \widetilde{Y}_t), \tag{4}$$

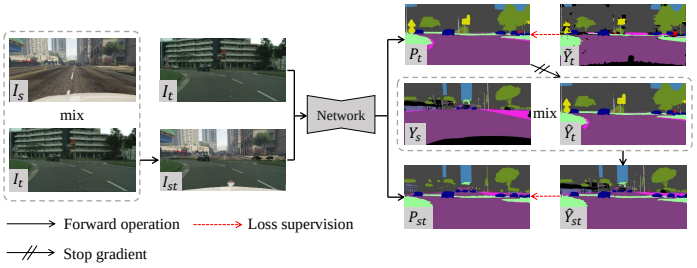

Fig. 4: Illustration of our proposed training process. Best viewed in color.

where L_{CE} is the cross-entropy loss. However, the independent supervision on source domain (*i.e.*, $L_{CE}(I_s, Y_s)$) inevitably brings about domain shift problem. Besides, in this work, the superpixels inherently contain noise, *i.e.*, some superpixels may contain pixels from more than one category, resulting in noisy active label $\widetilde{Y}_t$. To address these problems, our work proposes to utilize cross-domain mixing [14, 35] and pseudo label [25] with consistency regularization [26, 33] techniques. As shown in Fig. 4, we randomly select half of the classes in I_s and extract the corresponding pixels, which are then transplanted onto I_t, resulting in domain-mix sample I_{st}. The corresponding label $\hat{Y}_{st}$ is constructed by mixing Y_s and $\hat{Y}_t$ which is the pseudo label of I_t. Thus, the overall learning objective can be presented as follows:

$$L = L_{CE}(I_t, \widetilde{Y}_t) + L_{CE}(A(I_{st}), \hat{Y}_{st}), \tag{5}$$

where A denotes the data augmentations applied to the input I_{st}. In particular, we choose colorjitter, gaussianblur, and grayscale, as suggested in [26, 43]. Here, we employ pseudo label with consistency regularization in order to prevent overfitting to the label noise of $\widetilde{Y}_t$. We utilize data-augmentation-based consistency regularization to encourage model to generate consistent predictions for different variants of target data through data augmentation, thereby decreasing the likelihood of assigning wrong class labels [18, 43].

4 Experiments

Dataset. Following previous works [39, 40, 44], we evaluate various active learning and domain adaptation methods on two widely-used domain adaptive semantic segmentation benchmarks: GTA5→Cityscapes and SYNTHIA→Cityscapes. GTA5 [29] contains 24,966 1914×1052 synthesized images rendered by the gaming engine GTAV, acting as a source domain dataset. SYNTHIA [30] contains 9,400 1280×760 synthesized images, which serve as another source dataset. Cityscapes [10] is a representative dataset in the fields of semantic segmentation and autonomous driving domain. It comprises 2,975 images for training and 500 images for validation, both with a resolution of 2048×1024.

Table 1: Results on GTA5 → Cityscapes benchmark. "*" denotes the upper bound performance of superpixel-level methods, namely, the maximum model performance achievable by **labeling all superpixels**. We report the mIoU and best results are shown in **bold**.

Method	Architecture	Cost	road	side.	buil.	wall	fence	pole	light	sign	veg.	terr.	sky	pers.	rider	car	truck	bus	train	motor	bike	mIoU
Source Only	DeepLabv2 [5]	-	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
ADVENT [36]	DeepLabv2 [5]	-	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
DACS [35]	DeepLabv2 [5]	-	89.9	39.7	87.9	30.7	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	27.3	34.0	52.1
ProDA [45]	DeepLabv2 [5]	-	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
DAformer [14]	SegFormer [41]	-	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
LabOR [31]	DeepLabv2 [5]	40(0.008%)	96.1	71.8	88.8	47.0	46.5	42.2	**53.1**	60.6	89.4	55.1	91.4	70.8	44.7	90.6	56.7	47.9	39.1	47.3	62.7	63.5
RIPU [40]	DeepLabv2 [5]	40(0.008%)	95.5	69.2	88.2	**48.0**	46.5	36.9	45.2	55.7	88.5	**55.3**	90.2	69.2	46.1	91.2	70.7	73.0	58.2	50.1	65.9	65.5
ASAL [17]	DeepLabv2 [5]	40(0.008%)	94.0	67.6	86.5	34.4	44.2	42.8	51.7	62.5	87.7	47.2	89.5	69.8	48.6	87.4	55.7	68.8	55.1	50.9	64.0	63.6
Ours(Target-base)	DeepLabv2 [5]	23(0.004%)	96.5	73.2	87.5	41.5	45.6	37.1	47.9	63.6	88.4	41.5	**92.3**	73.4	47.2	92.1	71.1	74.6	63.8	51.6	60.0	65.7
Ours	DeepLabv2 [5]	40(0.008%)	**96.8**	**77.2**	**89.0**	38.4	**47.6**	**44.1**	52.9	**65.9**	**90.0**	50.9	91.4	**74.8**	**54.0**	**92.2**	**71.5**	**75.7**	**67.6**	**59.1**	**66.2**	**68.7**
LabOR [31]	DeepLabv2 [5]	11,535(2.2%)	96.6	77.0	89.6	47.8	50.7	48.0	56.6	63.5	89.5	57.8	91.6	72.0	47.3	91.7	62.1	61.9	48.9	47.9	65.3	66.6
UBD [44]	DeepLabv2 [5]	11,535(2.2%)	93.9	67.5	89.2	52.7	53.0	51.6	56.7	64.5	89.5	54.5	89.4	74.9	51.1	92.8	**75.4**	74.7	47.9	52.5	70.1	68.5
RIPU [40]	DeepLabv2 [5]	11,535(2.2%)	96.5	74.1	89.7	**53.1**	51.0	43.8	53.4	62.2	90.0	57.6	92.6	73.0	53.0	92.8	73.8	78.5	62.0	55.6	70.0	69.6
Ours	DeepLabv2 [5]	80(0.015%)	96.5	76.5	89.7	45.0	50.1	47.6	55.2	66.9	90.4	52.7	91.3	76.5	55.0	92.3	73.9	78.4	65.6	57.8	66.5	69.9
Ours	DeepLabv2 [5]	160(0.031%)	97.0	78.3	90.1	44.0	51.3	49.0	58.4	69.5	90.4	53.1	91.7	77.0	55.9	92.7	73.6	81.1	67.9	**58.2**	68.1	70.9
Ours	DeepLabv2 [5]	320(0.061%)	97.1	79.4	90.1	43.8	**53.4**	49.8	59.9	70.1	90.9	55.7	92.4	**77.8**	**57.4**	**92.9**	73.3	**83.2**	71.3	57.2	**71.0**	71.9
Ours	DeepLabv2 [5]	640(0.122%)	**97.4**	**80.2**	**90.6**	45.8	52.3	**52.7**	**61.5**	**71.6**	**91.1**	**58.2**	**93.1**	77.6	56.9	92.6	73.9	82.6	**73.4**	55.2	70.7	**72.5**
Ours*	DeepLabv2 [5]	873(0.167%)	97.3	80.4	90.7	47.1	52.7	52.9	62.3	72.1	91.2	57.9	93.0	77.4	57.7	92.8	74.3	81.4	69.3	58.2	71.3	72.6
Ours* (no fusion)	DeepLabv2 [5]	4,985(0.951%)	97.4	81.1	91.0	47.0	54.7	53.5	61.7	72.6	91.1	59.7	93.3	78.3	59.2	92.5	67.5	83.9	70.8	61.5	72.3	73.1
Fully Supervised	DeepLabv2 [5]	524,288(100%)	97.5	81.3	90.8	45.6	53.6	59.1	65.5	74.4	91.2	59.5	92.8	80.1	60.4	94.1	72.7	77.1	63.4	60.1	74.7	73.4
MADA [28]	DeepLabv3+ [6]	26,215(5%)	95.1	69.8	88.5	43.3	48.7	45.7	53.3	59.2	89.1	46.7	91.5	73.9	50.1	91.2	60.6	56.9	48.4	51.6	68.7	64.9
RIPU [40]	DeepLabv3+ [6]	26,215(5%)	97.0	77.3	90.4	**54.6**	53.2	47.7	55.9	64.1	90.2	**59.2**	**93.2**	75.0	54.8	92.7	73.0	79.7	68.9	55.5	70.3	71.2
D^{2}ADA [39]	DeepLabv3+ [6]	26,215(5%)	97.0	77.8	90.0	46.0	**55.0**	52.7	58.7	65.8	90.4	58.9	92.1	75.7	54.4	92.3	69.0	78.0	68.5	**59.1**	**72.3**	71.3
Ours	DeepLabv3+ [6]	640(0.122%)	**97.2**	**80.9**	**90.9**	48.1	**53.5**	**53.7**	**62.7**	**72.6**	**91.4**	58.3	92.8	**77.1**	**58.2**	**92.9**	**74.1**	**81.0**	**71.2**	58.5	70.6	**72.9**
Ours*	DeepLabv3+ [6]	873(0.167%)	97.4	81.1	91.0	47.9	53.8	53.6	63.5	72.8	91.3	58.6	92.9	77.3	58.4	92.7	74.4	81.2	71.7	58.8	71.1	73.1
Ours* (no fusion)	DeepLabv3+ [6]	4,985(0.951%)	97.4	81.2	91.1	47.9	55.3	57.1	62.1	72.4	91.2	59.9	92.6	78.8	58.1	93.4	75.1	81.1	71.9	58.1	73.1	73.6
Fully Supervised	DeepLabv3+ [6]	524,288(100%)	97.5	81.4	91.1	48.4	55.4	58.8	63.1	72.8	91.7	60.5	93.2	79.3	57.8	94.1	76.4	82.4	68.6	59.4	74.0	74.0
RIPU [40]	SegFormer [41]	26,215(5%)	**97.6**	**81.0**	91.1	50.7	**57.6**	**55.1**	60.5	69.1	91.4	**61.3**	**94.6**	76.3	52.7	**93.9**	**84.8**	81.0	64.9	58.4	**70.9**	73.3
Ours	SegFormer [41]	640(0.122%)	97.5	80.8	**91.2**	**53.3**	57.5	50.3	**60.9**	**71.8**	**91.6**	60.2	94.3	**76.5**	**56.4**	93.4	83.7	**85.4**	**76.2**	**61.5**	70.5	**74.4**
Ours*	SegFormer [41]	873(0.167%)	97.6	81.2	91.4	56.2	59.0	50.4	61.0	71.5	91.6	60.6	94.4	76.6	56.6	93.4	84.2	87.7	78.5	61.5	70.4	74.9
Ours*	SegFormer [41]	4,985(0.951%)	97.9	83.4	91.8	57.6	59.4	50.9	61.1	72.5	91.8	64.8	94.5	76.9	57.8	93.3	83.1	87.5	80.3	61.4	71.2	75.6
Fully Supervised	SegFormer [41]	524,288(100%)	98.0	84.2	92.4	60.6	58.2	60.9	66.5	76.1	92.3	65.7	94.7	79.6	60.6	94.6	84.1	85.0	68.4	60.2	74.7	76.7

Implementation details. All experiments are conducted on a single RTX 2080 Ti GPU with 12 GB memory. For fair and full comparisons, we choose three semantic segmentation architectures in our experiments, including two widely-used CNN-based models, DeepLabv2 [5] and DeepLabv3+ [6], and an excellent Transformer-based model, SegFormer [41]. For the CNN architecture, most hyper-parameters are kept identical to those used in [39, 40]. We use an SGD optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The learning rate is set at 2.5×10^{-4}, which is annealed following the poly learning rate policy with a power of 0.9. For the Transformer architecture, we follow the experimental setup of DAFormer [14]. In Fig. 3, we pre-train the SSN on source domain and follow the experimental setup in [15]. We use DAFormer [14] as UDA-merge to extract entropy map E_t. Notably, for accurately gauging domain differences, the UDA-base model necessitates an identical network architecture to our model (*i.e.* Target-base and Final model). Specifically, when utilizing CNN architecture, we adopt DACS [35] as the UDA-base method, whereas when using Transformer architecture, we use DAFormer [14] as our UDA-base method. For all experiments, the models are trained for $250,000$ iterations and early stopped at $120,000$ iteration with batches of 2, and all images are randomly cropped to 1024×512. The threshold hyperparameters τ in Eq. 1 is set as 0.05, and ε in Eq. 2 is set as 0.10, in line with the settings given by [17].

Annotation budget. Following RASL [3] and ASAL [17], we use the number of clicks as the labeling budget. Existing ADA methods set the budget as $11,535\,(2.2\%)$ or $26,215\,(5\%)$ pixels (*i.e.*, $11,535\,(2.2\%)$ or $26,215\,(5\%)$ clicks) per image, which is costly and impractical. In this work, we extract about $5,000$ superpixels per image and obtain around $1,000$ merged superpixels per image after low-uncertainty superpixel fusion. Consequently, more practical annotation budgets of 40, 80, 160, 320, and 640 clicks per image were selected for all experiments. Specifically, we label all MLUSPs in phase 1 and use the remaining budget to select HUSPs for annotations in phase 2, as in Fig. 3.

4.1 Performance Comparison

We compare our superpixel-level method with various domain adaptation methods, including unsupervised domain adaptation (UDA) [14,35,36,45] and active domain adaptation (ADA) [31,39,40,44]. Furthermore, we expand the application of ASAL [17] to active domain adaptation benchmarks, allowing for the comparison of various superpixel-level methods. The results on GTA5→Cityscapes are shown in Table 1, while the results on SYNTHIA→Cityscapes are provided in supplementary material. Notably, existing ADA methods have not conducted experiments on Transformers. For comprehensive experiments, we re-implement RIPU [40] on transformers for fair comparison. It can be seen that our method dramatically outperforms the previous method while greatly reducing the annotation costs.

GTA5→Cityscapes. In this scenario, we utilize SSN pretrained on GTA5 to extract approximately $4,985$ superpixels per image. Subsequently, through fusion using DAFormer trained on GTA5, we obtain around 873 superpixels. Compared with recent UDA methods such as ProDA [45] and DAFormer [14], our proposed method achieves a performance improvement of 15% and 6.2% respectively, with active-labeling only 640 clicks per image on target domain under the architectures of CNN and Transformer. Under a low-cost budget, our method achieves good results with only labeling all MLUSP (*i.e.*, 23 clicks per image). At the cost of 40, our method outperform RIPU [40] and ASAL [17] by 3.2% and 5.1% mIoU, respectively. Our proposed method is able to surpass RIPU [40], using only 80 clicks per image, as compared to RIPU's $11,535$ clicks. Furthermore, our method achieves a 1.6% higher mIoU compared to D^2ADA [39], which utilizes $26,215$ clicks, despite our usage of mere 640 clicks. Overall, our proposed superpixel-level method greatly reduced the annotation cost while improving the model performance on target domain. The experiments conducted on various architectures reveal that the performance of a model trained with complete annotation of merged superpixels (i.e., cost of 873) closely matches that of a model trained with complete annotation of original superpixels (i.e., cost of $4,985$). These results demonstrate the effectiveness of our proposed low-uncertainty superpixel fusion module in mitigating superpixel fusion errors, thus ensuring high-quality annotations.

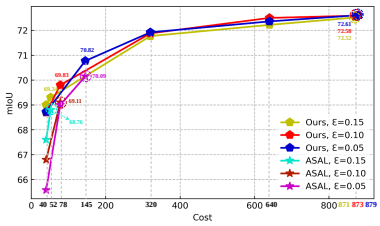

Fig. 5: Ablation study on various superpixel fusion methods.

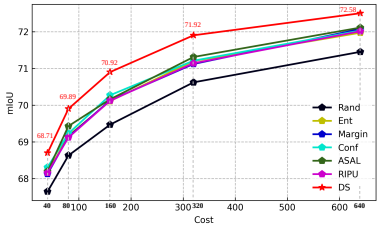

Fig. 6: Effect of selection algorithms used in $\mathcal{A}_2$.

4.2 Ablation Study

In this section, we conducted experiments to demonstrate the effectiveness of our proposed method. All experiments were conducted on the GTA5→Cityscapes using the DeepLabv2 backbone. Extensive experiments involving hyper-parameter settings, visual comparisons, and ablation studies on the SYNTHIA→Cityscapes are provided in the supplementary material.

Ablation on Superpixel-level Methods. A key of our proposed superpixel-level method is to mitigate fusion errors in domain adaptation using our low-uncertainty superpixel fusion module, as illustrated in Fig. 2(c). In this regard, we conducted experiments with various superpixel fusion methods while maintaining consistency in all other experimental settings. As depicted in Fig. 5, the point within the dotted circle represents the upper bound performance achieved by labeling all superpixels, and obviously, the upper bound of ASAL [17] is notably low. For instance, when setting the fusion threshold ε to 0.10, ASAL achieves a mere 69.11% mIoU by labeling all merged superpixels (at a cost of 78), which falls significantly short of the upper bound performance achieved by our method (72.58% mIoU). Furthermore, our method consistently outperforms ASAL even when employing identical annotation costs. It is evident that ASAL suffers from fusion errors caused by the inter-class feature confusion issue in domain adaptation. In contrast, our proposed low-uncertainty superpixel fusion module effectively addresses this problem, resulting in improved performance, especially the upper bound performance. Moreover, our method is not sensitive to hyperparameter ε because the LUSPs contain distinguishable features and thereby can be easily fused.

Ablation on Acquisition Functions. To demonstrate the superiority of our proposed acquisition functions, we make a comparison with other common strategies, *i.e.*, Random sampling (RAND), ASAL [17], and domain differences (DS). ASAL [17] considers both Margin [16] and class imbalance, and DS is the strategy we used as $\mathcal{A}_2$ at labeling phase 2. These common strategies are single-stage, which select and annotate samples from all superpixels according to their respective criteria. Unlike them, we use different strategies to select and annotate merged superpixels (MLUSPs) and unmerged superpixels (HUSPs). Specifically, our method adopts two-stage labeling and introduces two acquisition functions, namely $\mathcal{A}_1$ and $\mathcal{A}_2$, for labeling phases 1 and 2, respectively, as illustrated in

Table 2: Performance comparison using different acquisition strategies is conducted, where the acquisition function is denoted as AF.

Labeling phase	AF \ cost	40	80	160	320	640
single-stage	RAND	24.6	36.3	37.5	54.4	63.8
	ASAL	57.0	63.2	65.4	69.5	70.6
	DS	58.5	63.8	66.2	69.8	71.0
two-stage	**Ours**	**68.7**	**69.9**	**70.9**	**71.9**	**72.5**

Fig. 3. We first use $\mathcal{A}_1$ to select information-rich superpixels with large sizes, encompassing all MLUSPs. Then, we utilize $\mathcal{A}_2$ to select the remaining superpixels (HUSPs) with large domain differences (*i.e.*, DS). As shown in Table 2, our method consistently outperforms the other strategies across various cost settings. In Table 2, DS is equivalent to using only the second phase $\mathcal{A}_2$ in our method. However, DS overlooks the size of the superpixels and solely considers the domain differences among all superpixels, thus leading to inferior performance. This is because large-size superpixels often contain numerous pixels with small domain differences, causing them to be excluded by DS. Consequently, the number of annotated pixels solely marked by DS is significantly reduced. This highlights the crucial role of prioritizing information-rich superpixels with large sizes (i.e., MLUSPs) to achieve better performance. The supplementary material further demonstrates that prioritizing all MLUSPs is the optimal strategy.

After determining the prioritized annotation of MLUSPs in the first phase, we conducted experiments on the selection algorithm used at labeling phase 2 (*i.e.*, $\mathcal{A}_2$). Specifically, we compared ours (domain differences, DS) with other common selection methods such as RAND, entropy (ENT) [37], Margin [16],

Table 3: Ablation study of our proposed training techniques. Specifically, we verified the effectiveness of these techniques in different methods, including ours and RIPU [40].

				Ours(40)	Ours(640)	RIPU(11,535)
Method	A	B	C	mIoU	mIoU	mIoU
$\mathbf{M}^{(0)}$				63.9	**66.6**	62.8
$\mathbf{M}^{(1)}$	✓			67.4	**71.3**	68.2
$\mathbf{M}^{(2)}$	✓	✓		68.0	**71.7**	70.7
$\mathbf{M}^{(3)}$	✓	✓	✓	68.7	**72.5**	71.9

A: Cross-domain mixing technique
B: Pseudo label technique
C: Consistency regularization technique

softmax confidence (Conf) [11], ASAL [17] and RIPU [40]. RIPU utilizes region impurity and prediction uncertainty. As shown in Fig. 6, our DS selection algorithm surpassed the second-placed approach by 0.41% to 0.65% mIoU across different budget settings, whereas the gap between the second and sixth place was only 0.15% to 0.23%. Other uncertainty-based selection algorithms, which solely consider the difficult samples on target domain, fail to account for the challenging instances in domain adaptation scenarios. As a result, these methods only achieve sub-optimal performance. It is worth noting that the gain of our method is observed in the prioritized MLUSPs annotation in the first phase. The first phase of annotation completed around 82.03% of the pixel annotations. Therefore, based on the first-stage annotation, the performance differences between the selection methods in the second stage are not significant.

Ablation on Training Techniques. To verify the effectiveness of our proposed training techniques, we perform an ablation study with the following variants: $\mathbf{M}^{(0)}$: the baseline method in [39,44], *i.e.*, using all labeled data from source and target domain to train network; $\mathbf{M}^{(1)}$: extend $\mathbf{M}^{(0)}$ using cross-domain mixing technique to alleviate domain shift issue; $\mathbf{M}^{(2)}$: extend $\mathbf{M}^{(1)}$ by using pseudo labels on target domain; $\mathbf{M}^{(3)}$: extend $\mathbf{M}^{(2)}$ by using consistency regularization on domain-mix samples; As shown in Table 3, the effectiveness of each technique is demonstrated by the consistent improvements observed from $\mathbf{M}^{(0)}$ to $\mathbf{M}^{(3)}$ across three active learning settings. Furthermore, our proposed superpixel-level method outperforms RIPU with a significantly lower labeling cost of only 640, compared to RIPU's much higher cost of $11,535$, under identical training settings. This suggests that our superpixel-level method is capable of acquiring a substantial number of informative labeled pixels with minimal annotation costs.

5 Conclusion

In this paper, we have presented a novel approach to tackle the challenge of active domain adaptation in semantic segmentation. By introducing a superpixel-level approach, we have successfully reduced annotation costs while preserving model performance. Our proposed low-uncertainty superpixel fusion module effectively alleviates fusion errors that commonly arise in domain adaptation scenarios. Furthermore, we have designed two efficient acquisition functions for the comprehensive selection of information-rich superpixels. For model training, we propose to use cross-domain mixing and pseudo label with consistency regularization techniques respectively to address the domain shift and label noise problems. Extensive experiments on two challenging benchmarks demonstrate the effectiveness and efficiency of our approach, which outperforms existing state-of-the-art pixel-level methods by a large margin at 40x lower cost. We believe that our proposed superpixel-level method offers a promising new approach for active domain adaptation in semantic segmentation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 62176246. This work is also supported by Anhui Province Key Research and Development Plan (202304a05020045) and Anhui Province Natural Science Foundation (2208085UD17).

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. TPAMI (2012)
2. Van den Bergh, M., Boix, X., Roig, G., De Capitani, B., Van Gool, L.: Seeds: Superpixels extracted via energy-driven sampling. In: ECCV (2012)
3. Cai, L., Xu, X., Liew, J.H., Foo, C.S.: Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In: CVPR (2021)
4. Casanova, A., Pinheiro, P.O., Rostamzadeh, N., Pal, C.J.: Reinforced active learning for image segmentation. arXiv preprint arXiv:2002.06583 (2020)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2017)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
7. Chen, L., Wei, Z., Jin, X., Chen, H., Zheng, M., Chen, K., Jin, Y.: Deliberated domain bridging for domain adaptive semantic segmentation. NeurIPS (2022)
8. Cheng, Y., Wei, F., Bao, J., Chen, D., Wen, F., Zhang, W.: Dual path learning for domain adaptation of semantic segmentation. In: ICCV (2021)
9. Colling, P., Roese-Koerner, L., Gottschalk, H., Rottmann, M.: Metabox+: A new region based active learning method for semantic segmentation using priority maps. arXiv preprint arXiv:2010.01884 (2020)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
11. Culotta, A., McCallum, A.: Reducing labeling effort for structured prediction tasks. In: AAAI (2005)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
13. Gao, Y., Wang, Z., Zhuang, J., Zhang, Y., Li, J.: Exploit domain-robust optical flow in domain adaptive video semantic segmentation. In: AAAI (2023)
14. Hoyer, L., Dai, D., Van Gool, L.: Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: CVPR (2022)
15. Jampani, V., Sun, D., Liu, M.Y., Yang, M.H., Kautz, J.: Superpixel sampling networks. In: ECCV (2018)
16. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: CVPR. IEEE (2009)
17. Kim, H., Oh, M., Hwang, S., Kwak, S., Ok, J.: Adaptive superpixel for active learning in semantic segmentation. In: ICCV (2023)
18. Koh, K.B., Fernando, B.: Consistency regularization for domain adaptation. In: ECCV (2022)

19. Lee, S., Choi, W., Kim, C., Choi, M., Im, S.: Adas: A direct adaptation strategy for multi-target domain adaptive semantic segmentation. In: CVPR (2022)
20. Li, J., Wang, Z., Gao, Y., Hu, X.: Exploring high-quality target domain information for unsupervised domain adaptive semantic segmentation. In: ACM MM (2022)
21. Li, Z., Chen, J.: Superpixel segmentation using linear spectral clustering. In: CVPR (2015)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
23. Liu, Y., Deng, J., Gao, X., Li, W., Duan, L.: Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In: ICCV (2021)
24. Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., Rother, C.: Cereals-cost-effective region-based active learning for semantic segmentation. arXiv preprint arXiv:1810.09726 (2018)
25. Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: ECCV (2020)
26. Melas-Kyriazi, L., Manrai, A.K.: Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In: CVPR (2021)
27. Milioto, A., Stachniss, C.: Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using cnns. In: ICRA (2019)
28. Ning, M., Lu, D., Wei, D., Bian, C., Yuan, C., Yu, S., Ma, K., Zheng, Y.: Multi-anchor active domain adaptation for semantic segmentation. In: ICCV (2021)
29. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: ECCV. Springer (2016)
30. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
31. Shin, I., Kim, D.J., Cho, J.W., Woo, S., Park, K., Kweon, I.S.: Labor: Labeling only if required for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8588–8598 (2021)
32. Siam, M., Gamal, M., Abdel-Razek, M., Yogamani, S., Jagersand, M., Zhang, H.: A comparative study of real-time semantic segmentation for autonomous driving. In: CVPR workshops (2018)
33. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. NeurIPS (2020)
34. Sumithra, R., Suhil, M., Guru, D.: Segmentation and classification of skin lesions for disease diagnosis. Procedia Computer Science (2015)
35. Tranheden, W., Olsson, V., Pinto, J., Svensson, L.: Dacs: Domain adaptation via cross-domain mixed sampling. In: WACV (2021)
36. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR (2019)
37. Wang, D., Shang, Y.: A new active labeling method for deep learning. In: 2014 International joint conference on neural networks (IJCNN). IEEE (2014)
38. Wang, Q., Dai, D., Hoyer, L., Van Gool, L., Fink, O.: Domain adaptive semantic segmentation with self-supervised depth estimation. In: ICCV (2021)
39. Wu, T.H., Liou, Y.S., Yuan, S.J., Lee, H.Y., Chen, T.I., Huang, K.C., Hsu, W.H.: D 2 ada: Dynamic density-aware active domain adaptation for semantic segmentation. In: ECCV (2022)
40. Xie, B., Yuan, L., Li, S., Liu, C.H., Cheng, X.: Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In: CVPR (2022)

41. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. NeurIPS (2021)
42. Yang, F., Sun, Q., Jin, H., Zhou, Z.: Superpixel segmentation with fully convolutional networks. In: CVPR (2020)
43. Yang, L., Zhuo, W., Qi, L., Shi, Y., Gao, Y.: St++: Make self-training work better for semi-supervised semantic segmentation. In: CVPR (2022)
44. You, F., Li, J., Chen, Z., Zhu, L.: Pixel exclusion: Uncertainty-aware boundary discovery for active cross-domain semantic segmentation. In: ACM MM (2022)
45. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: CVPR (2021)
46. Zhang, Y., Wang, Z.: Joint adversarial learning for domain adaptation in semantic segmentation. In: AAAI (2020)