SEMANTIC SEGMENTATION BASED UNSUPERVISED DOMAIN ADAPTATION VIA PSEUDO-LABEL FUSION

ABSTRACT

In this paper, we propose a pseudo label fusion framework (PLF), a learning framework developed to deal with the domain gap between a source domain and a target domain for performing semantic segmentation based UDA in the unseen target domain. PLF fuses the pseudo labels generated by an ensemble of teacher models. The fused pseudo labels are then used by a student model to distill out the information embedded in these fused pseudo labels to perform semantic segmentation in the target domain. To examine the effectiveness of PLF, we perform a number of experiments on both GTA5→Cityscapes and SYNTHIA→Cityscapes benchmarks to quantitatively and qualitatively inspect the improvements achieved by employing PLF in performing semantic segmentation in the target domain. Moreover, we provide a number of parameter analyses to validate that the choices made in the design of PLF is both practical and beneficial. Our experimental results on both benchmarks shows that PLF indeed offers superior performance benefits in performing semantic segmentation in the unseen domain, and is able to achieve competitive performance when compared to the contemporary UDA techniques.

1 INTRODUCTION

In the past few years, semantic segmentation has been attracting the attention of computer vision researchers. Many supervised semantic segmentation methods have been proposed and achieved remarkable performance (Yu et al., 2017; Lin et al., 2017; Yu & Koltun, 2016; Badrinarayanan et al., 2017; Long et al., 2015; Yuan et al., 2019; Wu et al., 2019; Sandler et al., 2018; Chen et al., 2014; 2017a;b; 2018; Zhao et al., 2017). A number of them have even already been applied to real-life applications such as autonomous vehicles (Hong et al., 2018). However, supervised semantic segmentation methods typically require abundant labeled training data, which are usually expensive to annotate and are commonly unavailable in most real-world scenarios. Furthermore, models trained in simulated environments or other scenes often fail to generalize to the domain of deployment, especially when there exists a significant domain gap between the source and target domains (Tsai et al., 2018). To address this issue, unsupervised domain adaptation (UDA) methods have been introduced to bridge different domains (Ganin & Lempitsky, 2015; Liu et al., 2019a; Saito et al., 2018a; Tzeng et al., 2017; Zellinger et al., 2017; Long et al., 2018; Pinheiro, 2018; Shu et al., 2018; Saito et al., 2018b; Hoffman et al., 2016; 2018; Luo et al., 2019a; Gong et al., 2019; Dundar et al., 2020; Chen et al., 2019a; Wu et al., 2018; Tsai et al., 2018; Luo et al., 2019b; Chen et al., 2019b; Li et al., 2019; Du et al., 2019; Yang et al., 2020; Tsai et al., 2019; Vu et al., 2019a; Zhang et al., 2018; Chen et al., 2017c; Zheng & Yang, 2020a; Choi et al., 2019). UDA models learn to generalize to a target domain by training with the annotated data from a source domain and the unlabeled data from a target domain. If the domain gap can be effectively filled, intelligent machines such as autonomous vehicles and robots trained in virtual worlds can be transferred to the real world with relative ease.

A number of past endeavors have been dedicated to improving the performance of semantic segmentation based UDA through different approaches, and have achieved impressive results (Hoffman et al., 2016; 2018; Luo et al., 2019a; Gong et al., 2019; Wu et al., 2018; Tsai et al., 2018; Yang et al., 2020; Tsai et al., 2019; Vu et al., 2019a; Zhang et al., 2018; Chen et al., 2017c; Zheng & Yang, 2020a; Choi et al., 2019; Zou et al., 2018b;a; 2019; Zheng & Yang, 2020b; Tranheden et al., 2020; Luo et al., 2019b; Lee et al., 2018; Chen et al., 2019a; Vu et al., 2019b). The authors in (Luo et al., 2019b; Hoffman et al., 2016; 2018; Luo et al., 2019a; Gong et al., 2019; Wu et al., 2018; Tsai et al., 2018; Yang et al., 2020; Tsai et al., 2019; Vu et al., 2019a; Zhang et al., 2018; Chen et al., 2017c; Zheng & Yang, 2020a; Chen et al., 2019b; Li et al., 2019; Du et al., 2019) resorted to adversarial domain adaptation (ADA) methods, through which the domain discrepancy is minimized by training a generator and a discriminator against each other. These attempts were effective in bridging the domain gap as they demonstrated superior performance over models trained solely in their source domains. More recent works have opted for pseudo labelling and self-training frameworks (Choi et al., 2019; Zou et al., 2018b;a; 2019; Zheng & Yang, 2020b; Tranheden et al., 2020), which aim

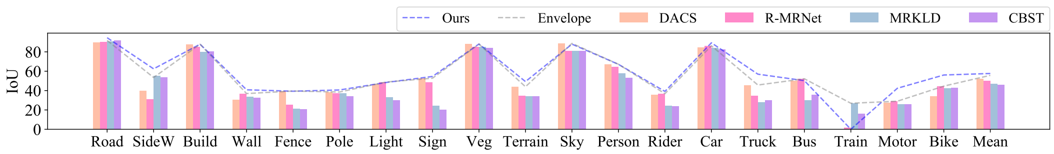

Figure 1: The distributions of the accuracy achieved by several semantic segmentation based UDA methods evaluated in Cityscapes. The IoUs of them are shown in colored bars, while the IoUs of their envelope and our method are plotted in gray and blue dashed lines, respectively. The mIoUs for the methods are denoted as 'Mean' in the figure, and are illustrated as the colored bars on the rightmost end.

to minimize the entropy of a model's predictions in the target domain. Previous works have shown that these self-training methods were able to outperform ADA methods by a considerable margin while maintaining their simplicity. Unfortunately, most of them learn from only one distribution of semantic classes produced by a single model, and the performance is therefore upper-bounded by the performance of that specific model, leaving space for improvements. Moreover, previous works have shown that with ensemble learning, the prediction quality can be improved if there exist significant differences between the decision boundaries of the models (Opitz & Maclin, 1999). Furthermore, it is observed that the per-class accuracy tend to vary by a substantial margin for different UDA training methods, even if their average accuracy is very similar, as shown in Fig. 1. This indicates that there might exist significant differences in the decision boundaries for different methods, making ensemble learning a promising and potentially effective approach for semantic segmentation based UDA. In light of these properties and the shortcomings of the prior methods, we argue that the performance of semantic segmentation based UDA can be further enhanced if ensemble learning is employed to incorporate knowledge from different models.

In this work, we propose an ensemble learning framework that distills the knowledge of an ensemble of teacher models to a single student model. Instead of being constrained by a teacher model, we introduce an aggregation procedure, called *pseudo label fusion (PLF)*, to approximate the ground truth distribution by leveraging the advantages of different teacher models in predicting different semantic classes. The dashed lines in Fig. 1 reveal that it is possible to derive a distribution that better interprets the target domain by PLF. Rather than directly using the fused PLs themselves as the prediction result, we employ the proposed PLF approach to train our student model to learn from the fused PLs (denoted as '*PLF (Ours)*' in Fig. 1). This enables us to effectively reduce the model size and the computational cost of the student, allowing it to be deployed in real-world applications. We evaluate our framework with two commonly-adopted metrics, GTA5 (Richter et al., 2016) to Cityscapes (Cordts et al., 2016) and SYNTHIA (Ros et al., 2016) to Cityscapes, and report the results both quantitatively and qualitatively against a number of baselines. We validate the generalizability of our framework by evaluating it on the test set of Cityscapes. In addition, we provide an analysis on the fusion methods, the threshold, and the backbone model used. Moreover, we demonstrate that our results are fully stable and reproducible. The primary contributions are thus summarized as follows:

- We introduce a framework utilizing ensemble learning and output-level knowledge distillation for enhancing the performance of semantic segmentation based UDA.
- We propose a pseudo label fusion procedure that utilizes the clustering property of semantic classes.
- We evaluate our framework under various configurations, and demonstrate its superior performance over the baselines in terms of its effectiveness and efficiency.

2 Related Works

2.1 Unsupervised Domain Adaptation

A number of methods has been proposed to bridge the domain gap between different domains (Tsai et al., 2018). There have been several approaches to this problem. One branch of these works adopted the ADA framework (Hoffman et al., 2016; 2018; Luo et al., 2019a; Gong et al., 2019; Dundar et al., 2020; Chen et al., 2019a; Wu et al., 2018; Tsai et al., 2018; Luo et al., 2019b; Chen et al., 2019b; Li et al., 2019; Du et al., 2019; Yang et al., 2020; Tsai et al., 2019; Vu et al., 2019a; Zhang et al., 2018; Chen et al., 2017c; Zheng & Yang, 2020a; Choi et al., 2019) to learn a representation of the target domain. These approaches enable significant improvements over those trained directly in their source domains. Due to the relatively higher training complexity and larger model sizes, recent researchers have turned their attention to self-training methods to tackle UDA tasks. These works utilize the technique of pseudo labeling (PL) along with techniques such as regularization (Zou et al., 2019; Zheng & Yang, 2020b), class-balancing (Zou et al., 2018a), data augmentations (Tranheden et al., 2020), and distillation (Tranheden et al., 2020) methods. The results presented in the literature demonstrated that the adoption of pseudo label self-training improves the performance significantly.

2.2 Pseudo Labeling

Pseudo labeling is a self-training method that was originally proposed to improve the performance of classification networks (Lee, 2013). It is accomplished by minimizing the entropy of model predictions, resulting in a better decision boundary. Subsequent works (Xie et al., 2020) have also shown promising results in image classification using pseudo labeling. Pseudo labeling was then further extended to the field of semantic segmentation (Yu et al., 2017; Lin et al., 2017; Yu & Koltun, 2016; Badrinarayanan et al., 2017; Long et al., 2015; Yuan et al., 2019; Wu et al., 2019; Sandler et al., 2018; Chen et al., 2014; 2017a;b; 2018; Zhao et al., 2017; Hung et al., 2018; Huang et al., 2018; Cholakkal et al., 2016; Kolesnikov & Lampert, 2016; Wei et al., 2017; Saleh et al., 2016; Shimoda & Yanai, 2016; Pinheiro & Collobert, 2015; Qi et al., 2016; Hong et al., 2016; Wei et al., 2016; Fan et al., 2020; Wang et al., 2020), and has gained successs by incorporating the information of the unlabeled data to improve performance. Since self-training via PL and UDA shares many similar characteristics in terms of problem formulation, it has been extensively used to solve UDA problems. In this paper, we mainly focus on semantic segmentation based UDA. The authors in (Zou et al., 2019; Zheng & Yang, 2020b; Zou et al., 2018a) showed promising results of semantic segmentation based UDA by PL. The authors in (Tranheden et al., 2020) further extended the fine tuning training procedure of the prior works and proposed a semi-supervised teacher-student training framework. They showed that their method could outperform their prior works when data augmentation techniques are incorporated.

2.3 Knowledge Distillation

The main idea of knowledge distillation is to use a smaller, faster model to learn knowledge from one or more teacher models Buciluǎ et al. (2006). The authors in Hinton et al. (2015) studied how training on a soft transfer set and alternation settings of the teacher model ensemble can improve the image classification performance of the student network. The authors in Cho & Hariharan (2019) investigated how a well-trained model may not be an ideal teacher model. They also elaborated on how the mismatch of the model sizes between the teachers and the student can potentially degrade the performance of the latter. Although knowledge distillation has been well-explored by prior endeavors in other domains such as Buciluǎ et al. (2006); Hinton et al. (2015); Cho & Hariharan (2019); Furlanello et al. (2018); Balan et al. (2015); Nguyen-Meidine et al. (2020a); Orbes-Arteainst et al. (2019); Liu et al. (2019b), the use of knowledge distillation in semantic segmentation based UDA remains relatively less explored. Although a few previous works Chen et al. (2019c); Nguyen-Meidine et al. (2020b); Gholami et al. (2020) have explored the use of knowledge distillation in multi-target UDA, single-source-single-target UDA methods utilizing knowledge distillation were still concentrated on the realm of image classification Nguyen-Meidine et al. (2020a); Ruder et al. (2017). The authors in Shen et al. (2019) devised framework that combines ensemble learning and knowledge distillation to solve the problem of computational cost incurred by ensemble learning. The authors in Zhai et al. (2020) adopted similar framework to solve person re-identification UDA problems. However, our work is different from Zhai et al. (2020); Shen et al. (2019) in terms of motivation and methodology. The emphasis of our work is dedicated to the adaptation of the output labels, where the information generated by the ensemble is interpretable. This enables the adoption of the proposed pseudo label fusion method, through which the advantages from the ensemble are combined, and the entropy minimization process can be accomplished.

3 Proposed Framework

In this section, we introduce the proposed framework. We first describe the problem definition. Then, we walk through the proposed method as well as the workflow of it. Finally, we explain the implementation details of PLF, which is designed to incorporate knowledge from multiple teachers.

3.1 Problem Definition

In general, the predictions of any semantic segmentation model can be interpreted as a distribution of semantic classes. Given a training procedure Γ, a model μ, and a set of input image-label pairs

$$\Omega^{\Gamma,\mu}_{\{X,Y\}} \rightarrow \Omega^{*} \quad (1) \quad \Omega^{\Gamma,\mu}_{\{X_{src},Y_{src},X_{tgt}\}} \rightarrow \Omega^{*}_{tgt} \quad (2) \quad \Omega^{fused} = \Lambda\left[\Omega^{\Gamma_i,T_i}_{\{X_{src},Y_{src},X_{tgt}\}}\right] \quad (3)$$

$$\Omega^{fused} \rightarrow \Omega^{*}_{tgt} \quad (4) \quad \Omega^{\Gamma_S,S}_{\{X_{tgt},Y^{fused}_{tgt}\}} \rightarrow \Omega^{fused} \quad (5) \quad \Omega^{\Gamma_S,S}_{\{X_{tgt},Y^{fused}_{tgt}\}} \rightarrow \Omega^{*}_{tgt} \quad (6)$$

$\{X, Y\}$, the distribution of the predictions from μ can be written as $\Omega^{\Gamma,\mu}_{\{X,Y\}}$. The objective of Γ is to guide $\Omega^{\Gamma,\mu}_{\{X,Y\}}$ to approach the distribution of the ground truth annotation Ω^{*}, formulated as Eq. (1). Under the context of UDA, Γ has access to the image-label pairs (X_{src}, Y_{src}) from a source domain, but only the images X_{tgt} from a target domain. The goal is to train a model μ through a training procedure Γ with $\{X_{src}, Y_{src}, X_{tgt}\}$, such that it can best approximate the distribution of ground truth annotation Ω^{*}_{tgt} in the target domain. Thus, the training objective can be described as Eq. (2).

If only a single model is used to approximate Ω^{*}_{tgt}, the approximation capability may be limited. In this work, Γ employs multiple pre-trained teachers $\{T\}$ and a single student S. $\{T\}$ can be trained by any arbitrary methods that do not include the information of the semantic labels in the target domain.

3.2 Proposed Method

In order to achieve the objective of Eq. (2), our method consists of two stages: fusion and distillation.

3.2.1 Fusion

Previous works have shown that, with ensemble learning, Ω^{*}_{tgt} can be better approximated if the predictions from multiple models are combined (Opitz & Maclin, 1999). The same concept can be extended to semantic segmentation based UDA, where a better distribution of semantic labels in the target domain can be obtained by fusing the predictions from different UDA models. Specifically, such a distribution, denoted as Ω^{fused}, can potentially be obtained by applying an aggregation function Λ to the distributions of the predictions from a set of UDA models $\{T_i\}_{i\in[i,N]}$, where N denotes the number of models. Each of these UDA models is separately trained with a corresponding training procedure in$\{\Gamma_i\}_{i\in[i,N]}$, using the annotated source domain data and unlabeled target domain images $\{X_{src}, Y_{src}, X_{tgt}\}$. Under such a formulation, the distribution of the predictions for a tuple $\{\Gamma_i, T_i\}$ is represented as $\Omega^{\Gamma_i,T_i}_{\{X_{src},Y_{src},X_{tgt}\}}$, and Ω^{fused} can therefore be formulated as Eq. (3). As the objective of Ω^{fused} is to approximate Ω^{*}_{tgt} in the target domain, it can be expressed as Eq. (4).

3.2.2 Distillation

To avoid using the entire ensemble in the target domain and to reduce the model size, Ω^{fused} is used to train a compact S. This is realized via a procedure Γ_S, which trains S by the fused prediction Y^{fused}_{tgt} for each X_{tgt}. It allows the distribution of the prediction of S to approach Ω^{fused} as Eq. (5).

Combining the objectives of the two stages, i.e., Eqs. (4) and (5), the overall objective of the proposed method can be expressed as Eq. (6). It implies that the student model is able to learn to approach the distribution of the annotation Ω^{*}_{tgt} by learning to approximate Ω^{fused}. In order to perform semantic segmentation based UDA, the fusion and the distillation methods are required to be properly defined.

3.2.3 Workflow

Fig. 2 (a) illustrates the workflow of our proposed framework, which is composed of two stages: the fusion and the distillation stages, highlighted in blue and purple, respectively. In the fusion stage, each teacher model in $\{T_i\}_{i\in[i,N]}$ performs semantic segmentation prediction on an RGB input image X_{tgt}, where the predicted confidence map (i.e., the predicted certainty channels of semantic classes) is denoted as $\{P_i\}_{i\in[i,N]}$. Then, $\{\Omega^{\Gamma_i,T_i}_{\{X_{src},Y_{src},X_{tgt}\}}\}_{i\in[1,N]}$, are aggregated by Λ, which is practically implemented by the proposed PLF process, to fuse $\{P_i\}_{i\in[i,N]}$ into a one-hot map Y^{fused}_{tgt}. Please note that there exist other possible implementations for Λ. Next, in the distillation stage, S is trained and updated by minimizing the cross-entropy loss L_S between Y^{fused}_{tgt} and the predicted certainty tensor P_S of S on X_{tgt}. The loss function L_S is formulated as $L_S = cross\_entropy(P_S, Y^{fused}_{tgt})$. For more details of the derivation procedure, please refer to the appendix at the end of the manuscript.

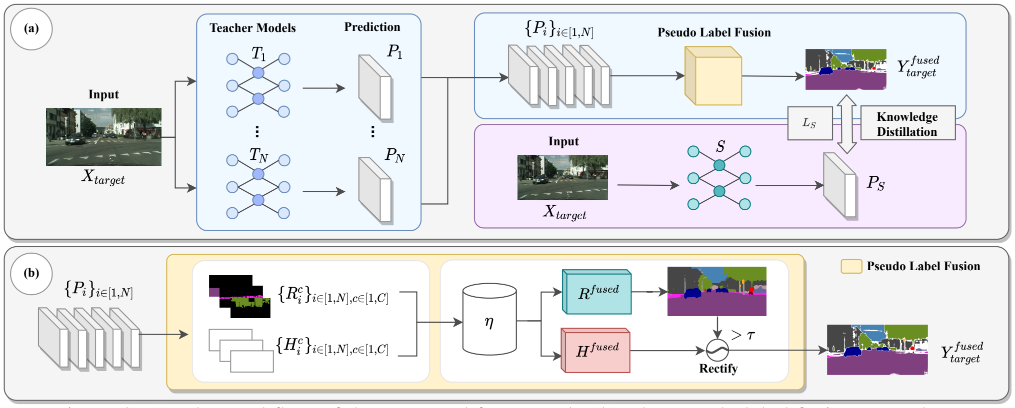

Figure 2: (a) The workflow of the proposed framework. (b) The pseudo label fusion procedure.

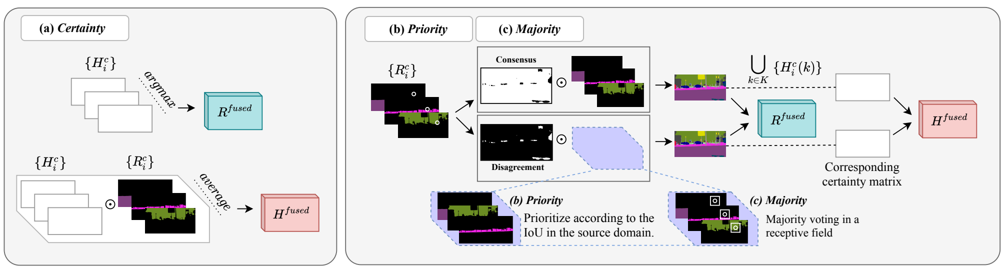

Figure 3: The three PLF procedures discussed in this work: (a) *certainty*, (b) *priority*, and (c) *majority*.

3.3 Pseudo Label Fusion Procedure

PLF as a whole can be viewed as a procedure that takes $\{P_i\}_{i\in[i,N]}$ as its input and outputs Y^{fused}_{tgt}. PLF is divided into three steps, the *preprocessing*, the *fusion*, and the *filtering* steps, as illustrated in Fig. 2 (b). In the first step, $\{P_i\}_{i\in[i,N]}$ is preprocessed on a per-class basis to separate and extract the per-class confidence map $\{H_i^c\}_{i\in[1,N],c\in[1,C]}$ and its corresponding PL $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$, where $c \in [1, C]$ and C is the number of classes. To carry the semantic knowledge of $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$ to S, in this work, we propose three different PLF processes: *certainty*, *priority*, and *majority* fusion. In the second step, $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$ are fused into one single PL by one of the proposed PLF processes $\eta = \{\eta_{certainty}, \eta_{priority}, \eta_{majority}\}$. The process η uses $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$ and $\{H_i^c\}_{i\in[1,N],c\in[1,C]}$ as the fusion criteria and produces the tuple (R^{fused}, H^{fused}), which represents the fused semantic PL and its corresponding confidence map. The last step involves filtering R^{fused} generated in the second step by applying a pixel-wise filtering threshold τ to remove the labels in R^{fused} whose certainty are below τ. The final result is a rectified PL Y^{fused}_{tgt}. The whole PLF procedure is formulated as:

$$\eta(\{H_i^c\}, \{R_i^c\}) = (H^{fused}, R^{fused}) \quad , where\ c \in [1, C]\ and\ i \in [1, N] \qquad (7)$$

$$Y^{fused}_{tgt} = \begin{cases} R^{fused}, & \text{if } H^{fused} > \tau \\ unlabeled, & \text{otherwise} \end{cases} \qquad (8)$$

The three fusion procedures are plotted in Fig. 3, and separately explained in the following sections.

3.3.1 Certainty Fusion

Certainty fusion employs an intuitive approach to integrate $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$ and $\{H_i^c\}_{i\in[1,N],c\in[1,C]}$, in which the labels with the highest certainty are chosen. It is accomplished by taking $argmax$ over c on $\{H_i^c\}_{i\in[1,N],c\in[1,C]}$ and max over c on $\{H_i^c\}_{i\in[1,N],c\in[1,C]}$, respectively, to obtain R^{fused} and H^{fused}, formulated as: $R^{fused} = argmax_c(H_i^c)$, $H^{fused} = \sum_{i=1}^{N} H_i^c \times R_i^c$.

3.3.2 Priority Fusion

Priority fusion aims to create a (R^{fused}, H^{fused}) tuple that is able to take the most advantages of the per-class IoU of $\{T_i\}_{i\in[1,N]}$. If there exists no disagreement among $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$ on a pixel, the value of R^{fused} for that pixel is determined by the consensus. However, if there is a disagreement on the same pixel among $\{R_i^c\}_{i\in[1,N],c\in[1,C]}$, the value of R^{fused} is decided according to the class IoU of the predictions of the teacher models. More specifically, if a pixel is labeled by two or more teacher models as different semantic classes, the teacher with the highest class IoU is chosen. As for H^{fused}, it is determined by $H^{fused} = \bigcup_{k \in K} H_i^c(k)$, where K is the pixels in the confidence map, k is the index of a pixel, and the values of c and i correspond to the chosen $R_i^c(k)$. $\bigcup$ denotes the union operator, and represents 'the collection of the corresponding certainty values from the teacher models.' Since the pseudo labels of different semantic classes are generated by the corresponding best-performing models, (R^{fused}, H^{fused}) is able to leverage the advantages provided by the ensemble $\{T_i\}_{i \in [1,N]}$.

3.3.3 Majority Fusion

Majority fusion targets at creating a (R^{fused}, H^{fused}) tuple using an approach based on majority voting that takes the surroundings of a pixel in to consideration. Similar to priority fusion, the value of a pixel in R^{fused} is determined by the consensus among $\{R_i^c\}_{i \in [1,N], c \in [1,C]}$ when all of them agree on the semantic prediction. If a disagreement is present among them, the value of that pixel is determined by performing majority voting on a fixed-sized receptive field around location where the disagreement happens. The same as the priority fusion procedure, in majority fusion, the confidence map H^{fused} is constructed by $H^{fused} = \bigcup_{k \in K} H_i^c(k)$, where the values of c and i correspond to the chosen $R_i^c(k)$.

4 Experimental Setup

4.1 Baselines and Evaluation Methods

We compare the results of our method against a number of baselines in terms of mIoU (mean intersection over union). The semantic segmentation based UDA baselines include AdaptSegNet (Tsai et al. (2018)), SIBAN (Luo et al. (2019a)), CLAN (Luo et al. (2019b)), APODA (Yang et al. (2020)), PatchAlign (Tsai et al. (2019)), AdvEnt (Vu et al. (2019a)), FDA-MBT (Yang & Soatto (2020)), PIT (Lv et al. (2020)), CBST (Zou et al. (2018b)), MRKLD (Zou et al. (2019)), MRNet (Zheng & Yang (2020a)), R-MRNet (Zheng & Yang (2020b)), and DACS (Tranheden et al. (2020)). We evaluate our method and the baselines on two commonly adopted benchmarks: GTA5 Richter et al. (2016)→Cityscapes Cordts et al. (2016) and SYNTHIA Ros et al. (2016)→Cityscapes. For the former, we train the models with 24,966 images-label pairs in the training set of GTA5 and 2,975 images in the training set of Cityscapes, and evaluate the per-class IoU of the nineteen semantic classes and the mIoU in the validation set of Cityscapes. For the latter, we train the models with 9,400 images-label pairs in the training set of SYNTHIA and 2,975 images in the training set of Cityscapes, and report the per-class IoU and mIoUs of 13 and 16 classes for comparing with different baselines.

4.2 Training Details

We adopt two widely-used semantic segmentation network architectures, Deeplabv2 (Chen et al. (2017a)) and Deeplabv3+ (Chen et al. (2018)), with three different backbones, ResNet-101 (**?**), DRN-D-54 (Yu et al. (2017)), and MobileNetV2 (Sandler et al. (2018)), as our student network. The teacher ensemble includes DACS, R-MRNe, MRKLD, CBST for GTA5→Cityscapes, and R-MRNet, MRKLD, CBST for SYNTHIA→Cityscapes. In the distillation stage, the student model is pretrained on the source domain and finetuned with the fused pseudo labels generated from the target domain. The student model is updated using SGD with learning rate 2.5×10^{-4} decreased with factor 0.9, weight decay 5×10^{-3}, momentum 0.9, and batch size 10 for 100K iterations for all the network architectures and backbones. The class-balancing strategy in CBST (Zou et al. (2018b)) is adopted during the training phase. For the proposed PLF method, the kernel size and the threshold (τ) in PLF are set to 5×5 and 0.9 respectively.

5 Experimental Results

5.1 A Comparison of Different Fusion Procedures

A comparison of the evaluation results of the three PLF procedures are summarized in Table 1 and Fig. 4 (a). These numerical and visualized results are obtained from the models trained in the GTA5 → Cityscapes task. The results indicate that priority fusion is able to out-perform the other two PLF methods in terms of mIoU. Albeit its superior performance, priority fusion requires additional prior knowledge in the form the per-class IoU of the teacher model in the validation set. At the same time, majority fusion offers comparable performance without these prior knowledge and is considered less artificial. Thus, in the rest sections, our results are presented based on the majority fusion procedure.

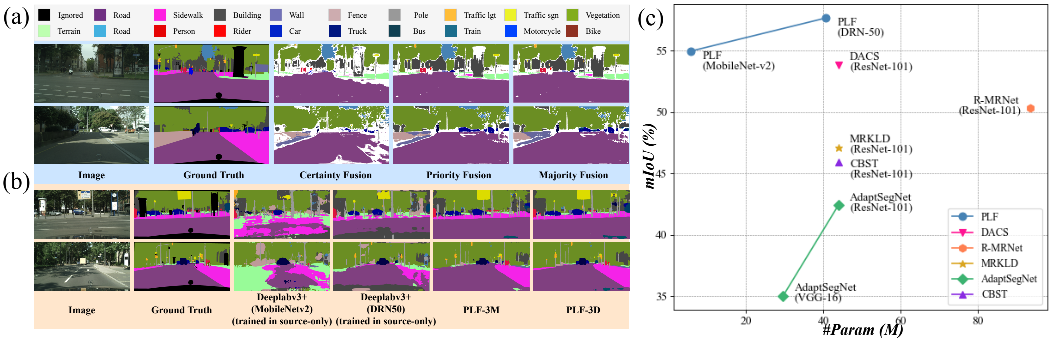

Figure 4: (a) Visualization of the fused PL with different PLF procedures. (b) Visualization of the model predictions, where PLF-3M and PLF-3D represent Deeplabv3+ trained with PLF using MobileNetv2 & DRN-50 as the backbones, respectively. (c) A comparison of mIoU v.s. model size.

Model	Backbone	#Param.	IS*	PLF	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
Deeplabv2	ResNet-101	43.9M	33.1 ms	Certainty	93.15	56.98	84.01	38.76	32.13	26.76	36.96	44.19	82.27	39.71	79.81	56.51	32.77	82.42	45.89	42.39	3.18	25.86	48.73	50.13
				Priority	93.44	56.96	84.74	40.76	37.70	27.47	38.18	46.33	83.73	45.15	82.76	56.91	33.66	84.19	54.60	51.74	0.00	36.64	51.21	52.96
				Majority	93.43	57.25	84.88	42.58	37.26	26.42	37.51	46.15	83.19	44.97	82.70	57.13	33.06	83.87	56.31	47.15	0.00	37.78	50.74	52.76
	DRN-D-54	35.6M	18.8 ms	Certainty	93.46	55.16	86.78	31.10	36.84	34.35	40.27	48.98	86.92	46.62	87.16	64.33	36.58	87.12	49.21	49.66	0.64	33.75	53.92	53.83
				Priority	93.81	59.33	86.79	28.60	37.21	34.36	41.83	50.00	87.09	46.42	87.60	63.81	36.06	87.13	51.76	53.12	1.06	33.67	53.07	54.35
				Majority	94.28	60.87	86.99	33.90	36.73	34.55	45.37	50.76	87.18	48.54	88.06	64.38	37.12	87.39	52.04	48.98	0.00	39.55	53.06	55.25
	MobileNetv2	2.0M	16.5 ms	Certainty	93.09	56.18	83.21	29.96	30.32	25.62	32.63	42.92	81.44	39.32	79.95	54.41	31.59	81.11	34.21	43.33	1.19	27.33	48.57	48.23
				Priority	93.27	55.75	83.86	35.04	37.56	26.21	35.30	45.31	82.78	42.48	81.89	55.56	32.16	83.05	45.87	50.40	1.01	37.44	50.76	51.35
				Majority	93.06	55.17	83.87	35.21	37.02	25.44	34.43	44.74	82.43	42.94	81.97	55.37	32.57	82.85	46.58	46.75	0.00	36.88	51.33	50.98
Deeplabv3+	ResNet-101	59.3M	35.1 ms	Certainty	93.89	59.98	86.07	30.60	32.02	37.92	43.18	49.77	85.66	39.00	84.89	62.51	33.52	85.17	41.61	41.66	3.92	24.89	51.23	51.97
				Priority	94.44	61.02	87.01	37.68	38.97	40.43	45.15	52.74	87.46	43.43	87.24	62.69	35.63	87.56	51.49	50.08	0.00	31.06	53.30	55.12
				Majority	94.38	61.77	86.78	39.86	38.04	38.37	43.96	52.93	86.92	43.50	86.80	62.25	35.01	86.37	53.14	44.99	0.00	32.51	52.60	54.75
	DRN-D-54	40.7M	22.1 ms	Certainty	93.97	59.43	87.50	31.98	34.33	41.33	47.86	52.06	86.47	40.79	86.12	66.19	37.89	87.54	45.60	49.50	0.17	38.33	55.12	54.85
				Priority	94.80	63.79	88.22	39.50	40.44	42.30	47.84	55.76	88.29	47.69	88.79	67.02	39.18	89.75	57.24	53.90	0.00	40.28	56.07	57.94
				Majority	94.62	62.46	87.90	40.81	39.34	40.62	48.36	54.58	88.12	49.29	87.96	67.12	39.14	89.34	56.95	50.24	0.03	42.37	56.09	57.65
	MobileNetv2	5.8M	20.9 ms	Certainty	93.85	58.68	85.84	32.79	33.33	35.62	40.72	48.38	85.52	40.31	84.36	60.96	33.15	85.67	35.88	44.68	3.74	26.96	50.64	51.64
				Priority	94.13	60.00	86.97	32.70	39.28	39.75	43.33	52.41	87.42	44.61	87.47	63.64	35.55	87.46	45.55	49.74	0.00	36.01	54.13	54.74
				Majority	94.29	61.15	87.19	35.94	40.63	38.59	43.35	51.41	87.32	44.81	87.71	64.12	35.57	87.77	46.60	47.32	0.00	36.67	53.56	54.95

*: The inference speed is evaluated based on an NVIDIA GTX TITAN V GPU

Table 1: A comparison of IoU and mIoU for the three PLF procedures with different configurations.

5.2 Comparisons of the Quantitative and Qualitative Results

Tables A5 compares the evaluation results of our method against the baselines based on the two metrics, respectively. The results of the baselines are presented in the upper rows of the tables, while ours and its variants are reported in the last few rows. It is observed that multiple reported per-class IoUs of our method, i.e., PFL-3D, are superior to those of baselines. It is also noted that our method PLF-3D outperforms previous works by a significant margin in terms of mIoU. Fig. 4 (c) plots the mIoU of some of the most representative baselines against the number of trainable parameters. It is observed that our framework is able to achieve superior performance even with lighter backbones containing less trainable parameters, e.g., MobileNetV2 and DRN-D-54. On the other hand, our method PFL-2R shows comparable results against the baselines if initialized from the pre-trained weights of Deeplabv2 (ResNet101) trained with supervision in the source domain. However, the performance is notably improved if initialized with the pre-trained weights of one of the UDA teacher models R-MRNet, i.e., PFL-2R (Ours†). Since our PLF method removes the uncertain predictions from the PLs, it excludes the knowledge in these low-certainty PLs during the distillation stage. As a result, the student model guesses the predictions of these pixels relying on the pre-trained weights. In contrast, since deeplabv2 (ResNet101) is only trained in source domain, it is unable to perform as well as R-MRNet in the target domain because of the existence of the domain gap, as mentioned in Section 1.

Fig. 4 (b) shows the visualized prediction results between Deeplabv3+ and our method. When compared with the prediction results of Deeplabv3+ trained solely in source domain, it is observed that the our PLF method is able to successfully perform UDA, as it is capable of correctly predicting the semantic labels of different objects with minimal noises. It is also observed that even with a compact model containing less parameters, the degradation of prediction accuracy is still tolerable.

Table 2 presents the test set prediction results of our framework. It is observed that our framework is still able to outperform those previously proposed UDA methods. Note that the results of DACS in Table 2 is a reproduced version since the statics on the test set is unavailable in the official release.

5.3 Analysis of the Backbone

Table. 1 reports the number of trainable parameters, the inference speed, and the semantic segmentation performance of the PLF framework with different student network backbones. It is observed

Method	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
R-MRNet	92.2	34.5	85.5	35.0	25.0	37.4	47.7	42.2	86.9	52.6	89.5	68.2	43.6	88.4	30.1	49.5	6.6	39.8	44.2	52.6
DACS	92.3	51.6	87.4	36.0	36.8	30.6	48.0	53.1	88.8	58.3	91.2	73.4	48.5	90.4	27.1	36.6	0.0	32.2	35.3	53.6
PLF-3D	94.8	59.1	88.2	35.9	37.4	38.0	47.5	52.1	88.9	57.9	90.6	72.3	47.0	91.2	49.9	50.7	3.08	44.1	51.0	57.9

Table 2: A comparison of R-MRNet, DACS, and our PLF-3D evaluated on the test set of Cityscapes.

GTA5 → Cityscapes

Method	Model (Backbone)	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
Source (Tsai et al. (2018))	Deeplabv2	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
AdaptSegNet (Tsai et al. (2018))	(ResNet-101)	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.2
SIBAN (Luo et al. (2019a))		88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
CLAN (Luo et al. (2019b))		87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
APODA (Yang et al. (2020))		85.6	32.8	79.0	29.5	25.5	26.8	34.6	19.9	83.7	40.6	77.9	59.2	28.3	84.6	34.6	49.2	8.0	32.6	39.6	45.9
PatchAlign (Tsai et al. (2019))		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
AdvEnt (Vu et al. (2019a))		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
FDA-MBT (Yang & Soatto (2020))		92.5	53.3	82.4	26.5	27.6	36.4	40.6	38.9	82.3	39.8	78.0	62.6	34.4	84.9	34.1	53.1	16.9	27.7	46.4	50.5
PIT (Lv et al. (2020))		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
Source (Zou et al. (2019))	Deeplabv2	71.3	19.2	69.1	18.4	10.0	35.7	27.3	6.8	79.6	24.8	72.1	57.6	19.5	55.5	15.5	15.1	11.7	21.1	12.0	33.8
CBST (Zou et al. (2018b))	(ResNet-101)	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRKLD (Zou et al. (2019))		91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	**26.9**	26.0	42.3	47.1
Source (Zheng & Yang (2020b))	Deeplabv2	51.1	18.3	75.8	18.8	16.8	34.7	36.3	27.2	80.0	23.3	64.9	59.2	19.3	74.6	26.7	13.8	0.1	32.4	34.0	37.2
MRNet (Zheng & Yang (2020a))	(ResNet-101)	89.1	23.9	82.2	19.5	20.1	33.5	42.2	39.1	85.3	33.7	76.4	60.2	33.7	86.0	36.1	43.3	5.9	22.8	30.8	45.5
R-MRNet (Zheng & Yang (2020b))		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
Source (Tranheden et al. (2020))	Deeplabv2	63.31	15.65	59.39	8.56	15.17	18.31	26.94	15.00	80.46	15.25	72.97	51.04	17.67	59.68	28.19	33.07	3.53	23.21	16.73	32.85
DACS (Tranheden et al. (2020))	(ResNet-101)	89.90	39.66	87.87	30.71	39.52	38.52	46.43	52.79	87.98	43.96	**88.76**	**67.20**	35.78	84.45	45.73	50.19	0.00	27.25	33.96	52.14
Source	Deeplabv2‡	75.76	18.06	70.94	17.75	13.53	14.04	15.84	6.35	78.57	21.30	76.08	44.76	5.64	69.40	19.15	24.10	0.00	4.09	0.77	30.32
PFL-2R (Ours)	(ResNet-101)	93.43	57.25	84.88	**42.58**	37.26	26.42	37.51	46.15	83.19	44.97	82.70	57.13	33.06	83.87	56.31	47.15	0.00	37.78	50.74	52.76
PLF-2R (Ours†)		94.16	59.88	87.47	41.50	39.85	36.44	46.87	54.27	86.92	46.99	86.54	65.28	38.84	88.52	**60.08**	**52.27**	0.00	**44.44**	55.58	57.15
Source	Deeplabv3+‡	21.13	7.47	51.42	8.15	10.11	20.31	20.83	14.97	70.94	4.93	64.77	37.58	7.07	51.51	12.07	9.69	9.85	3.56	15.16	23.24
PLF-3M (Ours)	(MobileNetV2)	94.29	61.15	87.19	35.94	**40.63**	38.59	43.35	51.41	87.32	44.81	87.71	64.12	35.57	87.77	46.60	47.32	0.00	36.67	53.56	54.95
Source	Deeplabv3+‡	57.40	21.43	56.80	8.93	22.14	32.38	34.62	24.90	78.98	15.92	63.71	55.55	13.83	58.11	21.99	29.78	2.36	28.41	33.98	34.80
PLF-3D (Ours)	(DRN-D-54)	**94.62**	**62.46**	**87.90**	40.81	39.34	**40.62**	**48.36**	**54.58**	**88.12**	**49.29**	87.96	67.12	**39.14**	**89.34**	56.95	50.24	0.03	42.37	**56.09**	**57.65**

SYNTHIA → Cityscapes

Method	Model (Backbone)	Road	SideW	Build	Wall*	Fence*	Pole*	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU	mIoU*
Source (Tsai et al. (2018))	Deeplabv2	55.6	23.8	74.6	-	-	-	6.1	12.1	74.8	-	79.0	55.3	19.1	39.6	-	23.3	-	13.7	25.0	-	38.6
AdaptSegNet (Tsai et al. (2018))	(ResNet-101)	84.3	42.7	77.5	-	-	-	4.7	7.0	77.9	-	82.5	54.3	21.0	72.3	-	32.2	-	18.9	32.3	-	46.7
SIBAN (Luo et al. (2019a))		82.5	24.0	79.4	-	-	-	16.5	12.7	79.2	-	82.8	58.3	18.0	79.3	-	25.3	-	17.6	25.9	-	46.3
CLAN (Luo et al. (2019b))		81.3	37.0	80.1	-	-	-	16.1	13.7	78.2	-	81.5	53.4	21.2	73.0	-	32.9	-	22.6	30.7	-	47.8
APODA (Yang et al. (2020))		86.4	41.3	79.3	-	-	-	22.6	17.3	80.3	-	81.6	56.9	21.0	84.1	-	49.1	-	24.6	45.7	-	53.1
PatchAlign (Tsai et al. (2019))		82.4	38.0	78.6	8.7	0.6	26.0	3.9	11.1	75.5	-	84.6	53.5	21.6	71.4	-	32.6	-	19.3	31.7	40.0	46.5
AdvEnt (Vu et al. (2019a))		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	48.0
FDA-MBT (Yang & Soatto (2020))		79.3	35.0	73.2	-	-	-	19.9	24.0	61.7	-	82.6	61.4	31.1	83.9	-	40.8	-	**38.4**	51.1	-	52.5
PIT (Lv et al. (2020))		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	51.8
Source (Zou et al. (2019))	Deeplabv2	64.3	21.3	73.1	2.4	1.1	31.4	7.0	27.7	63.1	-	67.6	42.2	19.9	73.1	-	15.3	-	10.5	28.9	34.9	40.3
CBST (Zou et al. (2018b))	(ResNet-101)	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	-	78.3	60.6	28.3	81.6	-	23.5	-	18.8	39.8	42.6	48.9
MRKLD (Zou et al. (2019))		67.7	32.2	73.9	10.7	1.6	**37.4**	22.2	31.2	80.8	-	80.5	60.8	29.1	82.8	-	25.0	-	19.4	45.3	43.8	50.1
Source (Zheng & Yang (2020b))	Deeplabv2	44.0	19.3	70.9	8.7	0.8	28.2	16.1	16.7	79.8	-	81.4	57.8	19.2	46.9	-	17.2	-	12.0	43.8	35.2	40.4
MRNet (Zheng & Yang (2020a))	(ResNet-101)	82.0	36.5	80.4	4.2	0.4	33.7	18.0	13.4	81.1	-	80.8	61.3	21.7	84.4	-	32.4	-	14.8	45.7	43.2	50.2
R-MRNet (Zheng & Yang (2020b))		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	54.9
Source (Tranheden et al. (2020))	Deeplabv2	36.30	14.64	68.78	9.17	0.20	24.39	5.59	9.05	68.96	-	79.38	52.45	11.34	49.77	-	9.53	-	11.03	20.66	29.45	33.65
DACS (Tranheden et al. (2020))	(ResNet-101)	80.56	25.12	81.90	**21.46**	2.85	37.20	22.67	23.99	83.69	-	**90.77**	**67.61**	**38.33**	82.92	-	38.90	-	28.49	47.58	48.34	54.81
Source	Deeplabv2‡	65.39	21.88	65.80	6.52	0.28	18.48	5.70	16.33	73.22	-	69.80	41.39	12.95	70.08	-	24.90	-	2.72	25.61	32.57	38.14
PLF-2R (Ours)	(ResNet-101)	87.80	43.43	81.16	18.96	3.75	26.30	28.56	34.00	80.77	-	82.71	55.53	19.55	83.48	-	46.69	-	24.76	49.37	47.93	55.22
PLF-2R (Ours†)		87.79	41.92	82.70	14.19	1.47	36.43	32.62	18.80	82.01	-	81.32	64.08	24.42	86.43	-	40.78	-	26.01	**54.25**	48.45	55.63
Source	Deeplabv3+‡	37.46	15.36	65.74	3.51	0.00	22.62	0.00	0.00	75.81	-	82.65	36.40	2.70	46.44	-	21.69	-	0.14	0.09	25.66	29.58
PLF-3M (Ours)	(MobileNetV2)	88.28	**47.35**	82.73	18.71	**5.13**	33.76	30.56	36.98	83.35	-	88.30	59.56	20.78	86.88	-	49.03	-	20.31	52.77	50.28	57.45
Source	Deeplabv3+‡	25.40	15.55	59.70	18.07	0.66	26.35	19.36	30.22	72.50	-	74.28	48.11	13.67	74.62	-	36.94	-	13.92	36.45	35.36	40.06
PLF-3D (Ours)	(DRN-D-54)	**88.38**	45.74	**83.45**	18.09	3.35	36.67	**34.99**	**38.12**	**84.65**	-	88.16	61.01	22.29	**87.64**	-	**53.28**	-	28.41	53.68	**51.76**	**59.22**

‡: Please note that the backbone of the teacher models for ‘PLF-2R’, ‘PLF-3M’ and ‘PLF-3D’ are all Deeplabv2 (ResNet-101).

Table 3: The experimental results evaluated on the GTA5→Cityscapes and SYNTHIA→Cityscapes benchmarks. The numbers presented in the middle and the last two columns correspond to per-class IoUs, mIoU, and mIoU*, respectively. mIoU* represents the average IoU over all the semantic classes excluding those with superscript *, and is adopted by a few baseline methods. ‘Source’ denotes that the student models only trained in the source domain. ‘Ours’ refers to the setting that the student models are pretrained in the source domain. ‘Ours†’ represents the evaluation setting in which the student model in our framework is initialized with the pretrained weights from R-MRNet Zheng & Yang (2020b).

that the student models with ResNet-101 as the backbone are clearly under-performing, despite of its large model capacity, while those with DRN-D-54 achieve superior performance. The results also reveal that little performance is sacrificed if MobileNetv2, whose model size is significantly smaller than the other two, is used. Due to the small model size, it is able to perform inference at the speed of around 20 milliseconds, which translates to 50 fps, making it deployable for real-time applications.

5.4 Analysis of the Threshold

Fig. 5 demonstrates the how the filter threshold τ affects the performance of the student model. In the proposed framework, pixels of the fused PL with certainty below the filter threshold τ is removed in order to improve the quality of the PLs as well as the performances of the student models. As illustrated in Fig. 5, the performances of PLF-3D and PLF-3M increase as the filter threshold τ

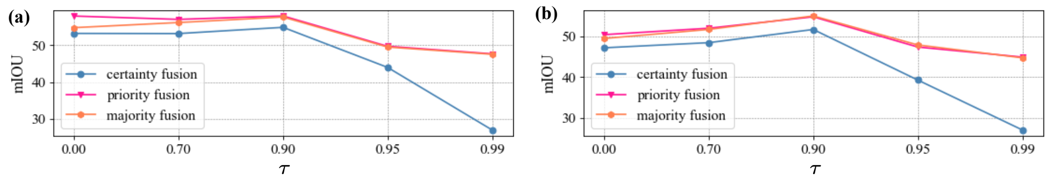

Figure 5: Comparisons of (a) τ v.s. mIoU for PLF-3D, and (b) τ v.s. mIoU for PLF-3M.

	CBST	MRKLD	R-MRNet	DACS	mIoU
PLF-3M	√	√			48.05
	√	√	√		50.25
	√	√	√	√	54.95
PLF-3D	√	√			50.09
	√	√	√		53.15
	√	√	√	√	57.65

Table 4: The performance (mIoU) comparison of our framework with different sets of teacher models on GTA5→Cityscapes.

increases. Nevertheless, this increasing trend stops at a certain point, i.e., 0.9 in our case, and the performance of the student networks fall drastically if the threshold is further increased. This may be due to the fact that if the filter threshold is too large, a large number of the labels in the fused PLs will be removed, implying that the amount of knowledge available for the student model to learn is reduced as well. Due to the lack of information in the PLs, the student models might be underfitted.

5.5 Ablation Study

The key motivation behind our learning framework is that different training methods may lead to different decision boundaries among the teacher models, which can be potentially utilized by the ensemble learning framework to improve the performance. To validate this concept, we perform experiments on different sets of teacher models and report the performance of our framework with different fusion functions. As shown in Table 4, when more teacher models trained with different methods are included in the ensemble, the higher the performance of our framework achieves. The experimental results also reveal that our design possesses the potential to evolve with time, since a newly proposed UDA training method can be added into the ensemble and further increases the performance of our framework.

6 Conclusion

In this paper, we proposed PLF, a learning framework developed to deal with the domain gap between a source domain and a target domain for performing semantic segmentation based UDA in the unseen target domain. In order to validate the proposed framework, we examined PLF as well as its variants, and compared them with the other recent UDA approaches on both the GTA5→Cityscapes and SYNTHIA→Cityscapes benchmarks. In our experiments, the proposed framework was able to outperform the baselines. Furthermore, we performed several parameter analyses, and investigated how different design choices may influence the performance of the proposed framework. As the technique that fuses pseudo labels from a teacher ensemble has been validated to be effective by our work. PLF thus pioneered a new direction for future semantic segmentation based UDA researches.

References

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, Jan. 2017.

A. K. Balan, V. Rathod, K. P. Murphy, and M. Welling. Bayesian dark knowledge. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3438–3446, Dec. 2015.

C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data mining (KDD-06)*, pp. 535–541, Aug. 2006.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *arXiv:1412.7062*, Jul. 2014.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, Apr. 2017a.

L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, Dec. 2017b.

L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 801–818, Sep. 2018.

Y. Chen, W. Li, X. Chen, and L. V. Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1841–1850, Jun. 2019a.

Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1791–1800, Jun. 2019b.

Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 1992–2001, Oct. 2017c.

Z. Chen, J. Zhuang, X. Liang, and L. Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2248–2257, Jun. 2019c.

J. H. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 4794–4802, Oct. 2019.

J. Choi, T. Kim, and C. Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 6830–6840, Oct. 2019.

H. Cholakkal, J. Johnson, and D. Rajan. Backtracking scspm image classifier for weakly supervised top-down saliency. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 5278–5287, Jun. 2016.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 982–991, Oct. 2019.

A. Dundar, M.-Y. Liu, Z. Yu, T.-C. Wang, J. Zedlewski, and J. Kautz. Domain stylization: A fast covariance matching framework towards domain adaptation. *IEEE Computer Architecture Letters*, (01):1–1, 2020.

J. Fan, Z. Zhang, C. Song, and T. Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 4283–4292, Jun. 2020.

T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born again neural networks. *Proc. Int. Conf. Machine Learning (ICML)*, Jul. 2018.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1180–1189, Jul. 2015.

B. Gholami, P. Sahu, O. Rudovic, K. Bousmalis, and V. Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29: 3993–4002, 2020.

R. Gong, W. Li, Y. Chen, and L. V. Gool. Dlow: Domain flow for adaptation and generalization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2477–2486, Jun. 2019.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, Mar. 2015.

J. Hoffman, D. Wang, F. Yu, and T. Darrell. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:1612.02649*, Dec. 2016.

J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proc. Int. Conf. Machine Learning (ICML)*, pp. 1989–1998, Jul. 2018.

S. Hong, J. Oh, H. Lee, and B. Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 3204–3212, Jun. 2016.

Zhang-Wei Hong, Chen Yu-Ming, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, Hsuan-Kung Yang, Brian Hsi-Lin Ho, Chih-Chieh Tu, Yueh-Chuan Chang, Tsu-Ching Hsiao, et al. Virtual-to-real: Learning to control in visual semantic segmentation. *Int. Joint Conf. Artificial Intelligence (IJCAI)*, Jul. 2018.

Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 7014–7023, Jun. 2018.

W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang. Adversarial learning for semi-supervised semantic segmentation. *BMVC 2018*, Sep. 2018.

A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 695–711, Oct. 2016.

D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, Jun. 2013.

K.-H. Lee, G. Ros, J. Li, and A. Gaidon. Spigan: Privileged adversarial learning from simulation. *arXiv:1810.03756*, 2018.

Y. Li, L. Yuan, and N. Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 6936–6945, Jun. 2019.

G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1925–1934, Jul. 2017.

X. Liu, S. Li, L. Kong, W. Xie, P. Jia, J. You, and B. Kumar. Feature-level frankenstein: Eliminating variations for discriminative recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 637–646, Jun. 2019a.

Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang. Structured knowledge distillation for semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2604–2613, Jun. 2019b.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Jun. 2015.

M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1640–1650, Dec. 2018.

Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 6778–6787, Oct. 2019a.

Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2507–2516, 2019b.

F. Lv, T. Liang, X. Chen, and G. Lin. Cross-domain semantic segmentation via domain-invariant interactive relation transfer. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4334–4343, 2020.

L. T. Nguyen-Meidine, E. Granger, M. Kiran, J. Dolz, and L.-A. Blais-Morin. Joint progressive knowledge distillation and unsupervised domain adaptation. *arXiv:2005.07839*, May 2020a.

L. T. Nguyen-Meidine, M. Kiran, J. Dolz, E. Granger, A. Bela, and L.-A. Blais-Morin. Unsupervised multi-target domain adaptation through knowledge distillation. *arXiv:2007.07077*, Jul. 2020b.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

M. Orbes-Arteainst, J. Cardoso, L. Sørensen, C. Igel, S. Ourselin, M. Modat, M. Nielsen, and A. Pai. Knowledge distillation for semi-supervised domain adaptation. In *OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging*, pp. 68–76. 2019.

P. O. Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 8004–8013, Jun. 2018.

P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1713–1721, Jun. 2015.

X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia. Augmented feedback in semantic segmentation under image level supervision. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 90–105, Oct. 2016.

S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 102–118, Oct. 2016.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 3234–3243, Jun. 2016.

S. Ruder, P. Ghaffari, and J. G Breslin. Knowledge adaptation: Teaching to adapt. *arXiv:1702.02052*, 2017.

K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Adversarial dropout regularization. In *Proc. Int. Conf. Learning Representations (ICLR)*, Apr. 2018a.

K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 3723–3732, Jun. 2018b.

F. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 413–432, Oct. 2016.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, Jun. 2018.

Z. Shen, Z. He, and X. Xue. Meal: Multi-model ensemble via adversarial learning. In *Proc. of the AAAI Conf. on Artificial Intelligence*, volume 33, pp. 4886–4893, 2019.

W. Shimoda and K. Yanai. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 218–234, Oct. 2016.

R. Shu, H. H. Bui, H. Narui, and S. Ermon. A dirt-t approach to unsupervised domain adaptation. In *Proc. Int. Conf. Learning Representations (ICLR)*, Apr. 2018.

W. Tranheden, V. Olsson, J. Pinto, and L. Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. *arXiv:2007.08702*, Jul. 2020.

Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 7472–7481, Jun. 2018.

Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker. Domain adaptation for structured output via discriminative patch representations. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 1456–1465, Oct. 2019.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 7167–7176, Jul. 2017.

T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2517–2526, Jun. 2019a.

T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 7364–7373, 2019b.

Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *roc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 12275–12284, Jun. 2020.

Y. Wei, X. Liang, Y. Chen, Y. Jie, Z.and Xiao, Y. Zhao, and S. Yan. Learning to segment with image-level annotations. *Pattern Recognition*, 59:234–244, 2016.

Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 39(11):2314–2320, Nov. 2017.

Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, and L. S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 518–534, Sep. 2018.

Z. Wu, C. Shen, and A. Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

Q. Xie, M.-T. Luong, E. Hovy, and Q. Le. Self-training with noisy student improves imagenet classification. In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 10687–10698, Jun. 2020.

J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, and L. Lin. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In *Proc. the Thirty-Fourth AAAI Conf. Artificial Intelligence (AAAI-20)*, pp. 12613–12620, Feb. 2020.

Y. Yang and S. Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4085–4095, 2020.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *Proc. Int. Conf. Learning Representations (ICLR)*, May 2016.

F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 472–480, Jul. 2017.

Y. Yuan, X. Chen, and J. Wang. Object-contextual representations for semantic segmentation. In *ECCV 2020 Spotlight*, Aug. 2019.

W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *Proc. Int. Conf. Learning Representations (ICLR)*, Apr. 2017.

Yunpeng Zhai, Qixiang Ye, Shijian Lu, Mengxi Jia, Rongrong Ji, and Yonghong Tian. Multiple expert brainstorming for domain adaptive person re-identification. *arXiv preprint arXiv:2007.01546*, 2020.

Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 6810–6818, Jun. 2018.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, Jul. 2017.

Z. Zheng and Y. Yang. Unsupervised scene adaptation with memory regularization in vivo. *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, Jul. 2020a.

Z. Zheng and Y. Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *arXiv:2003.03773*, Apr. 2020b.

Y. Zou, Z. Yu, B. Kumar, and J. Wang. Domain adaptation for semantic segmentation via class-balanced self-training. In *Proc. European Conf. Computer Vision (ECCV)*, Sep. 2018a.

Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proc. European Conf. Computer Vision (ECCV)*, pp. 289–305, Sep. 2018b.

Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang. Confidence regularized self-training. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 5982–5991, Oct. 2019.

Appendices

A1 Background Material

In this section, we review the background knowledge of the previous adversarial domain adaptation (ADA) methods and pseudo labeling (PL).

A1.1 Adversarial Domain Adaptation

For UDA problems, we have access to the image-label pairs (X_{src}, Y_{src}) in the source domain and the images X_{tgt} in the target domain, where $X_{src}, X_{tgt} \in \mathbb{R}^{H\times W\times 3}$, $Y_{tgt} \in \mathbb{R}^{H\times W\times C}$. The goal is to train a network to minimize the discrepancy between the semantic segmentation prediction and the target domain ground truth. Take AdaptSegNet as a example, a generator G and a discriminator D are trained against each other. The training objective of D is to distinguish whether the semantic segmentation outputs of G belongs to the source domain or not. On the other hand, the training objectives of the G is to confuse the discriminator D with its prediction. The loss function is defined as follows:

$$L_{G\_adv} = -\sum_{h,w} log(D(P_{tgt}^{(h,w,1)})), \tag{A1}$$

$$L_{D\_adv} = -\sum_{h,w} (1-z)log(D(P^{(h,w,0)})) + z(log(D(P^{(h,w,1)}))), \tag{A2}$$

where the segmentation softmax output of G is defined as $P_{src} = G(X_{src}^{(h,w,c)})$ in the source domain and $P_{tgt} = G(X_{tgt}^{(h,w,c)})$ in the target domain, $where P \in \mathbb{R}^{H\times W\times C}$. z is zero if the sample is drawn from the target domain, and is one if the sample is from the source domain. The cross-entropy loss between P_{src} and the ground truth in the source domain is also imposed to ensure that G preserves the general representation of the source domain. The source domain loss function can be defined as follows:

$$L_{G\_seg} = -\sum_{h,w}\sum_{c\in[1,C]} Y_{src}^{(h,w,c)} log(P_{src}^{(h,w,c)}). \tag{A3}$$

Algorithm 1: PLF Algorithm

```
While (S is not converged) do
    // generate PL from the teacher models.
    for i = 1 to N do
        P_i = T_i(X_target)
    end for
    {H_c^i},{R_c^i}  ← Select PL and their corresponding confidence maps from {P_i}.
    (H^fused, R^fused) = η({H_c^i},{R_c^i})
    Y_target^fused = (H^fused > τ)?  R^fused : unlabeled
    // distill the knowledge to the student model.
    P_S = S(X_target)
    L_S = cross_entropy(P_S, Y_target^fused)
    Compute ∇L_S by back-propagation and update the parameters in S.
end while
```

Figure A1: The pseudo code of the proposed PLF framework

A1.2 Pseudo Label

First proposed by (Lee, 2013), it was first intoduced to improve the performance of image classification by training with both the labeled data and unlabeled data. It is used in the fine-tunning phase of training during which the network is trained on both the supervised and unsupervised data. The unsupervised loss is calculated using the cross entropy between the network prediction and the pseudo labels, which is generated using the prediction on the unlabeled data with pixel-wise function described by the following equation:

$$Y_i^{'} = \begin{cases} 1, & i = argmax_{i'} f_{i'}(x) \\ 0, & \text{otherwise,} \end{cases} \tag{A4}$$

where $i, i\prime$ denotes one of the classes in a set of classes. $f_{i\prime}(x)$ denotes a pixel in the output confidence map. By reducing the cross entropy loss between the unlabeled data and the generated pseudo label, the class overlapping of the output is greatly reduced and the decision boundary is adjusted to lie in low density regions.

A2 Additional Details of the Proposed PLF Framework

In this section, we provide the pseudo code of our PLF framework in Section A2.1, detailed definition of the loss function in section A2.2, and the detailed training settings required to reproduce our results in Section A2.3.

A2.1 Pseudo Code

The PLF training framework is presented as pseudo code, and is presented in Algorithm 1. A batch of images X_{target} in the target domain is fed into a set of the teacher models T_i, and generates the prediction P_i. Then Y_{target}^{fused} is produced by the PLF process. Next, the student model generates the semantic prediction P_S, and the cross-entropy loss between Y_{target}^{fused} and P_S is imposed to distill the knowledge in the PLs.

A2.2 Loss Function

In the PLF framework, the fused PL Y_{target}^{fused} and the prediction P_S from the student model can be viewed as 3 dimensional tensors with size $H \times W \times C$, i.e., $Y_{target}^{fused}, P_S \in R^{(H \times W \times C)}$. The cross-entropy loss L_S between the two can be written as:

$$L_S = -\sum_{h,w} \sum_{c \in [1,C]} Y_{target}^{fused\,(h,w,c)} log(P_S^{(h,w,c)}). \tag{A5}$$

Table A1: The evaluation results on the validation set of Cityscapes.

Method	Backbone	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
Source	Deeplabv2	75.76	18.06	70.94	17.75	13.53	14.04	15.84	6.35	78.57	21.30	76.08	44.76	5.64	69.40	19.15	24.10	0.00	4.09	0.77	30.32
PLF-2R	(ResNet-101)	93.43	57.25	84.88	42.58	37.26	26.42	37.51	46.15	83.19	44.97	82.70	57.13	33.06	83.87	56.31	47.15	0.00	37.78	50.74	52.76
PLF-2R-T		94.16	59.88	87.47	41.50	39.85	36.44	46.87	54.27	86.92	46.99	86.54	65.28	38.84	88.52	60.08	52.27	0.00	44.44	55.58	57.15
Oracle		96.16	73.35	86.57	43.34	47.62	29.58	42.07	51.20	86.28	53.42	87.73	62.34	41.23	87.15	71.80	71.49	49.20	46.34	61.55	62.54
Source	Deeplabv3+	21.13	7.47	51.42	8.15	10.11	20.31	20.83	14.97	70.94	4.93	64.77	37.58	7.07	51.51	12.07	9.69	9.85	3.56	15.16	23.24
PLF-3M	(MobileNet)	94.29	61.15	87.19	35.94	40.63	38.59	43.35	51.41	87.32	44.81	87.71	64.12	35.57	87.77	46.60	47.32	0.00	36.67	53.56	54.95
Oracle		96.84	75.88	87.52	44.55	45.98	45.11	47.79	61.38	88.72	53.98	89.95	67.35	42.71	91.29	61.13	72.51	58.89	43.34	64.81	65.25
Source	Deeplabv3+	57.40	21.43	56.80	8.93	22.14	32.38	34.62	24.90	78.98	15.92	63.71	55.55	13.83	58.11	21.99	29.78	2.36	28.41	33.98	34.80
PLF-3D	(DRN-50)	94.62	62.46	87.90	40.81	39.34	40.62	48.36	54.58	88.12	49.29	87.96	67.12	39.14	89.34	56.95	50.24	0.03	42.37	56.09	57.65
Oracle		97.68	81.31	90.77	49.18	51.30	56.41	61.15	71.49	91.13	59.10	93.70	76.68	52.55	93.59	76.66	79.92	63.58	55.41	72.47	72.32

Table A2: An analysis of the thresholds and the evaluation results on the validation set of Cityscapes.

Backbone	Threshold	mIOU (certainty)	mIOU (priority)	mIOU (majority)
ResNet101	0.00	47.71	52.06	49.76
	0.70	49.14	52.67	50.57
	0.90	51.97	54.94	55.12
	0.95	42.46	49.03	48.71
	0.99	28.34	47.21	46.32
DRN-50	0.00	53.20	57.91	54.76
	0.70	53.16	57.01	56.15
	0.90	54.85	57.94	57.65
	0.95	43.97	49.70	49.50
	0.99	26.98	47.65	47.53
MobileNetV2	0.00	47.16	50.38	49.44
	0.70	48.42	51.95	51.65
	0.90	51.64	54.74	54.95
	0.95	39.27	47.35	47.86
	0.99	27.07	44.89	44.71

A2.3 The Detailed Training Settings

To detailed training setting used to produce the results of our experiment is shown as follows:

- Learning Rate: $2.5 * 10^{-4}$ with decay=0.9 (SGD)
- Weight Decay: 0.0005
- Momentum: 0.9
- 512*1024 $\rightarrow$ crop 256*512
- Batch Size: 10
- Iterations: 100K

A3 Additional Experimental Results

In this section we provide additional experimental results including, the performance of the student model trained with supervision in Cityscapes in section A3.1, the detailed experimental result with altered fusion threshold in Section A3.2, results that shows the reproducibility of our framework in Section A3.2.1, and additional visualization of the prediction in Section A3.3.

A3.1 Training Results in the Target Domain

Table A1 shows the performance of the student model when the annotations in the target domain is available. Specifically, the student model is trained under the same training procedure proposed in Section 4.2 but Y_{target}^{fused} is replaced by the ground truth Y_{target} of the target domain.

A3.2 Additional Results of the Threshold Analysis

Table A2 shows the performance results of our method with ResNet-101, DRN, and MobileNetv2 as the backbones along with different fusion filtering thresholds.

A3.2.1 Reproducibility

Table A3 and A4 demonstrate the reproducibility and the stability of the proposed method. Each row in the table represents a experiment which is run 5 times under the same experimental setting. The reported data only shows slight fluctuation in performance, indicating that the result of the proposed methodology have small variance and is therefore considered stable and reproducible, given the aforementioned experimental setups and hyperparameters. Furthermore, it is ensured

Table A3: The evaluation results of PLF on the validation set of Cityscapes.

	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
PLF-2R	93.36± 0.10	56.08± 1.15	84.89± 0.10	41.56± 0.75	37.00± 0.45	26.63± 0.11	37.76± 0.15	46.61± 0.10	83.29± 0.07	45.15± 0.34	82.43± 0.30	56.74± 0.21	33.41± 0.38	83.74± 0.10	55.29± 1.25	47.06± 0.46	0.00± 0.00	36.46± 0.46	50.72± 0.54	52.54± 0.24
PLF-3D	94.50± 0.22	61.58± 1.55	87.91± 0.15	35.87± 0.85	39.68± 0.89	40.74± 0.35	48.90± 0.67	55.13± 0.44	88.20± 0.05	48.93± 0.47	88.57± 0.39	67.06± 0.53	38.78± 1.12	89.26± 0.20	55.00± 2.74	50.48± 1.25	0.02± 0.06	40.03± 0.95	54.91± 1.20	57.13± 0.28
PLF-3M	94.22± 0.06	60.07± 1.05	87.10± 0.17	34.48± 1.19	38.75± 1.21	38.55± 0.32	43.57± 0.43	52.16± 0.79	87.24± 0.12	44.44± 1.11	87.24± 0.45	63.71± 0.26	35.43± 0.42	87.62± 0.86	46.97± 2.02	46.71± 1.18	0.00± 0.00	34.79± 3.60	53.74± 3.43	54.57± 0.45

Table A4: The evaluation results of PLF on the validation set of Cityscapes.

	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Sky	Person	Rider	Car	Bus	Motor	Bike	mIOU
PLF-2R	87.83± 0.04	43.42± 0.31	81.17± 0.11	18.85± 0.37	3.69± 0.29	26.07± 0.10	27.65± 0.86	34.05± 0.27	80.78± 0.10	82.60± 0.19	54.82± 0.33	18.78± 0.16	83.63± 0.16	46.09± 1.38	20.08± 0.64	49.05± 0.21	47.41± 0.15
PLF-3D	88.64± 0.19	47.04± 0.36	83.59± 0.08	19.43± 0.39	3.03± 0.31	36.11± 0.14	32.15± 2.57	37.87± 0.29	84.39± 0.35	87.56± 0.44	63.35± 0.41	21.12± 0.58	87.94± 0.20	52.58± 1.10	21.93± 1.97	53.76± 0.80	51.28± 0.13
PLF-3M	88.72± 0.18	46.91± 0.35	82.90± 0.16	18.68± 0.53	3.89± 0.16	34.40± 0.24	29.61± 1.18	36.93± 0.15	84.13± 0.17	88.25± 0.13	60.18± 0.24	19.35± 0.23	87.01± 0.24	49.01± 1.67	16.05± 2.66	52.30± 0.15	49.89± 0.26

that the source codes of the proposed framework are well verified and fully reproducible. For more details about the provided source codes, please refer to the anonymous GitHub repository https://anonymous.4open.science/r/8645c8cd-7baf-4adc-ab61-1ee908063a00/.

A3.3 VISUALIZATION

Fig. A2 shows additional visualization data that demonstrate the effectiveness of the proposed framework. The first two columns show the RGB in put image of Cityscapes and the ground truth. Column 3 and 4 shows the fused PLs and the prediction of the student model that corresponds to the input image displayed in column 1, respectively

A3.4 COMPLETE QUANTITATIVE RESULTS

Tables A5 and A6 show the complete comparison of the performance between the proposed method and the baselines in GTA5→Cityscapes and SYNTHIA→Cityscapes, respectively.

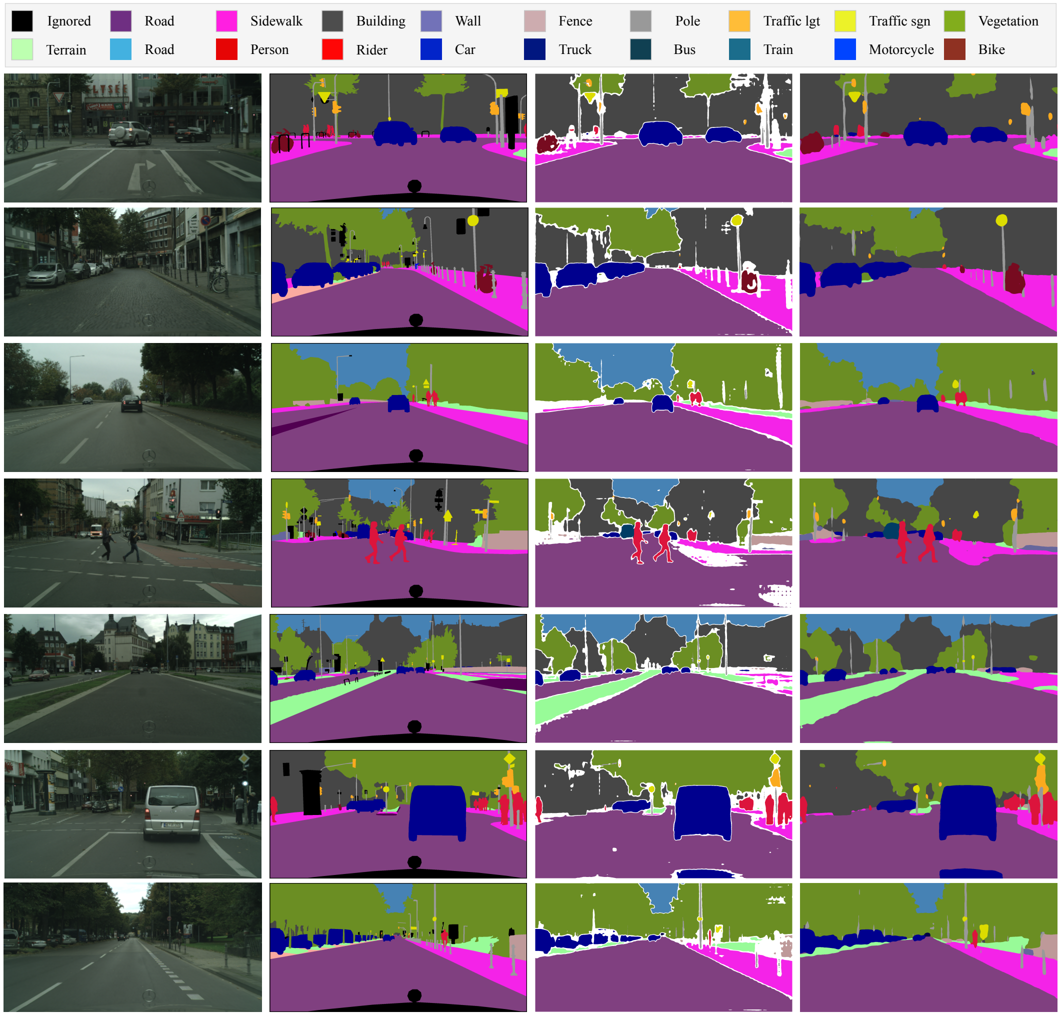

Image **Ground Truth** **Majority Fusion** **PLF-3D**

Figure A2: The visualization results of PLF-3D on Cityscapes.

Table A5: The evaluation results of the validation set on Cityscapes (GTA5→Cityscapes).

Method	Backbone	Road	SideW	Build	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bike	mIOU
Source	DRN-26	42.7	26.3	51.7	5.5	6.8	13.8	23.6	6.9	75.5	11.5	36.8	49.3	0.9	46.7	3.4	5.0	0.0	5.0	1.4	21.7
CyCADA		79.1	33.1	77.9	23.4	17.3	32.1	33.3	31.8	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5
Source	DRN-105	36.4	14.2	67.4	16.4	12.0	20.1	8.7	0.7	69.8	13.3	56.9	37.0	0.4	53.6	10.6	3.2	0.2	0.9	0.0	22.2
MCD		90.3	31.0	78.5	19.7	17.3	28.6	30.9	16.1	83.7	30.0	69.1	58.5	19.6	81.5	23.8	30.0	5.7	25.7	14.3	39.7
Source	Deeplabv2 (ResNet-101)	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
AdaptSegNet		86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.2
SIBAN		88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
CLAN		87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
APODA		85.6	32.8	79.0	29.5	25.5	26.8	34.6	19.9	83.7	40.6	77.9	59.2	28.3	84.6	34.6	49.2	8.0	32.6	39.6	45.9
PatchAlign		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
AdvEnt	Deeplabv2	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
Source	Deeplabv2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	29.2
FCAN	(ResNet-101)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46.6
Source	Deeplabv2 (ResNet-101)	71.3	19.2	69.1	18.4	10.0	35.7	27.3	6.8	79.6	24.8	72.1	57.6	19.5	55.5	15.5	15.1	11.7	21.1	12.0	33.8
CBST		91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRKLD		91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	**26.9**	26.0	42.3	47.1
MRKLD-SP-MST	ResNet38	91.7	45.1	80.9	29.0	23.4	43.8	47.1	40.9	84.0	20.0	60.6	64.0	31.9	85.8	39.5	48.7	25.0	38.0	47.0	49.8
Source	Deeplabv2 (ResNet-101)	51.1	18.3	75.8	18.8	16.8	34.7	36.3	27.2	80.0	23.3	64.9	59.2	19.3	74.6	26.7	13.8	0.1	32.4	34.0	37.2
MRNet		89.1	23.9	82.2	19.5	20.1	33.5	42.2	39.1	85.3	33.7	76.4	60.2	33.7	86.0	36.1	43.3	5.9	22.8	30.8	45.5
R-MRNet		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
Source	Deeplabv2	63.31	15.65	59.39	8.56	15.17	18.31	26.94	15.00	80.46	15.25	72.97	51.04	17.67	59.68	28.19	33.07	3.53	23.21	16.73	32.85
DACS	(ResNet-101)	89.90 / -	39.66 / -	87.87/ -	30.71 / -	39.52 / -	38.52 / -	46.43 / -	52.79 / -	87.98 / -	43.96 / -	**88.76/ -**	**67.20 / -**	35.78 / -	84.45 / -	45.73 / -	50.19 / -	0.00 / -	27.25 / -	33.96 / -	52.14 / 53.84
Source	Deeplabv2 (ResNet-101)	75.76	18.06	70.94	17.75	13.53	14.04	15.84	6.35	78.57	21.30	76.08	44.76	5.64	69.40	19.15	24.10	0.00	4.09	0.77	30.32
MMD-S-2R		93.43	57.25	84.88	**42.58**	37.26	26.42	37.51	46.15	83.19	44.97	82.70	57.13	33.06	83.87	56.31	47.15	0.00	37.78	50.74	52.76
MMD-T-2R		94.16	59.88	87.47	41.50	39.85	36.44	46.87	54.27	86.92	46.99	86.54	65.28	38.84	88.52	**60.08**	52.27	0.00	**44.44**	55.58	57.15
Source	Deeplabv3+	21.13	7.47	51.42	8.15	10.11	20.31	20.83	14.97	70.94	4.93	64.77	37.58	7.07	51.51	12.07	9.69	9.85	3.56	15.16	23.24
MMD-S-3M	(MobileNet)	94.29	61.15	87.19	35.94	**40.63**	38.59	43.35	51.41	87.32	44.81	87.71	64.12	35.57	87.77	46.60	47.32	0.00	36.67	53.56	54.95
Source	Deeplabv3+	57.40	21.43	56.80	8.93	22.14	32.38	34.62	24.90	78.98	15.92	63.71	55.55	13.83	58.11	21.99	29.78	2.36	28.41	33.98	34.80
MMD-S-3D	(DRN-50)	**94.62**	**62.46**	**87.90**	40.81	39.34	**40.62**	**48.36**	**54.58**	**88.12**	**49.29**	87.96	67.12	**39.14**	**89.34**	56.95	**50.24**	0.03	42.37	**56.09**	**57.65**

Table A6: The evaluation results of the validation set on Cityscapes (SYNTHIA→Cityscapes).

Method	Backbone	Road	SideW	Build	Wall*	Fence*	Pole*	Light	Sign	Veg	Sky	Person	Rider	Car	Bus	Motor	Bike	mIOU*	mIOU
Source	DRN-105	14.9	11.4	58.7	1.9	0.0	24.1	1.2	6.0	68.8	76.0	54.3	7.1	34.2	15.0	0.8	0.0	26.8	23.4
MCD		84.8	43.6	79.0	3.9	0.2	29.1	7.2	5.5	83.8	83.1	51.0	11.7	79.9	27.2	6.2	0.0	43.5	37.3
Source	Deeplabv2 (ResNet-101)	55.6	23.8	74.6	-	-	-	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6	-
AdaptSegNet		84.3	42.7	77.5	-	-	-	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7	-
SIBAN		82.5	24.0	79.4	-	-	-	16.5	12.7	79.2	82.8	58.3	18.0	79.3	25.3	17.6	25.9	46.3	-
CLAN		81.3	37.0	80.1	-	-	-	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	47.8	-
APODA		86.4	41.3	79.3	-	-	-	22.6	17.3	80.3	81.6	56.9	21.0	84.1	49.1	24.6	45.7	53.1	-
PatchAlign		82.4	38.0	78.6	8.7	0.6	26.0	3.9	11.1	75.5	84.6	53.5	21.6	71.4	32.6	19.3	31.7	46.5	40.0
AdvEnt		85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	48.0	41.2
Source	Deeplabv2 (ResNet-101)	64.3	21.3	73.1	2.4	1.1	31.4	7.0	27.7	63.1	67.6	42.2	19.9	73.1	15.3	10.5	28.9	40.3	34.9
CBST		68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	48.9	42.6
MRKLD		67.7	32.2	73.9	10.7	1.6	**37.4**	22.2	31.2	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	50.1	43.8
Source	Deeplabv2 (ResNet-101)	44.0	19.3	70.9	8.7	0.8	28.2	16.1	16.7	79.8	81.4	57.8	19.2	46.9	17.2	12.0	43.8	40.4	35.2
MRNet		82.0	36.5	80.4	4.2	0.4	33.7	18.0	13.4	81.1	80.8	61.3	21.7	84.4	32.4	14.8	45.7	50.2	43.2
R-MRNet		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	54.9	47.9
Source	Deeplabv2	36.30	14.64	68.78	9.17	0.20	24.39	5.59	9.05	68.96	79.38	52.45	11.34	49.77	9.53	11.03	20.66	33.65	29.45
DACS	(ResNet-101)	80.56 / -	25.12 / -	81.90 / -	**21.46/ -**	2.85 / -	37.20 / -	22.67 / -	23.99 / -	83.69 / -	**90.77/ -**	**67.61/ -**	**38.33/ -**	82.92 / -	38.90 / -	**28.49 / -**	47.58 / -	54.81 / 55.98	48.34 / 49.10
Source	Deeplabv2 (ResNet-101)	65.39	21.88	65.80	6.52	0.28	18.48	5.70	16.33	73.22	69.80	41.39	12.95	70.08	24.90	2.72	25.61	38.14	32.57
MMD-S-2R		87.80	43.43	81.16	18.96	3.75	26.30	28.56	34.00	80.77	82.71	55.53	19.55	83.48	46.69	24.76	49.37	55.22	47.93
MMD-T-2R		87.79	41.92	82.70	14.19	1.47	36.43	32.62	18.80	82.01	81.32	64.08	24.42	86.43	40.78	26.01	**54.25**	55.63	48.45
Source	Deeplabv3+	37.46	15.36	65.74	3.51	0.00	22.62	0.00	0.00	75.81	82.65	36.40	2.70	46.44	21.69	0.14	0.09	29.58	25.66
MMD-S-3M	(MobileNet)	88.28	**47.35**	82.73	18.71	**5.13**	33.76	30.56	36.98	83.35	88.30	59.56	20.78	86.88	49.03	20.31	52.77	57.45	50.28
Source	Deeplabv3+	25.40	15.55	59.70	18.07	0.66	26.35	19.36	30.22	72.50	74.28	48.11	13.67	74.62	36.94	13.92	36.45	40.06	35.36
MMD-S-3D	(DRN-50)	**88.38**	45.74	**83.45**	18.09	3.35	36.67	**34.99**	**38.12**	**84.65**	88.16	61.01	22.29	**87.64**	**53.28**	28.41	53.68	**59.22**	**51.76**