Union-set Multi-source Model Adaptation for Semantic Segmentation*

Zongyao Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama

Hokkaido University, Japan
{li,togo,ogawa,mhaseyama}@lmd.ist.hokudai.ac.jp

Abstract. This paper solves a generalized version of the problem of multi-source model adaptation for semantic segmentation. Model adaptation is proposed as a new domain adaptation problem which requires access to a pre-trained model instead of data for the source domain. A general multi-source setting of model adaptation assumes strictly that each source domain shares a common label space with the target domain. As a relaxation, we allow the label space of each source domain to be a subset of that of the target domain and require the union of the source-domain label spaces to be equal to the target-domain label space. For the new setting named union-set multi-source model adaptation, we propose a method with a novel learning strategy named model-invariant feature learning, which takes full advantage of the diverse characteristics of the source-domain models, thereby improving the generalization in the target domain. We conduct extensive experiments in various adaptation settings to show the superiority of our method.

Keywords: Model adaptation, domain adaptation, semantic segmentation

1 Introduction

Learning with unlabeled data is a long-term problem in the field of machine learning, and it has shown great significance more than ever since the rise of deep learning which heavily relies on well-labeled data. Due to the difficulty of unsupervised learning, some studies import labeled data from a different domain, and the problem accordingly becomes unsupervised domain adaptation (UDA) [9,19]. Typically, a method of UDA borrows the knowledge from a labeled source domain for learning with an unlabeled target domain by using data of both the domains and reducing the domain gap [14]. However, considering the fact that regulations are increasingly and strictly constituted for protecting private data, sharing the source-domain data may be impractical in some applications. For such situations, an alternative way for borrowing the source-domain knowledge without direct use of the source-domain data is necessary.

* This study was partly supported by JSPS KAKENHI Grant Number JP21H03456 and conducted on the Data Science Computing System of Education and Research Center for Mathematical and Data Science, Hokkaido University.

Model adaptation [15] (also named source-free domain adaptation) has been proposed for solving the above problem. As a derivative of UDA, model adaptation replaces the source-domain data with a source-domain pre-trained model and is thus no longer limited by the restriction of data sharing. Since private information can be hardly recovered from the pre-trained models, model adaptation faces less limitations and is practical in a wider range of applications than traditional UDA. Furthermore, in addition to the less difficulty of getting the access permission, pre-trained models also require considerably less storage size than training data, and therefore using multiple pre-trained models of different source domain, i.e., multi-source model adaptation (MSMA), seems to be a cost-efficient option when multiple appropriate source domains exist. However, unlike multi-source UDA which has been widely studied [35,36,11], the study on MSMA is insufficient despite its promising prospect. To the best of our knowledge, only one work studied on MSMA for image classification [1].

This paper focuses on the problem of MSMA for semantic segmentation which still remains to be solved. Similar to multi-source UDA, in a general MSMA setting, a common label space is expected to be shared by all the source domains and the target domain, which is a too stringent assumption to be practical in some real-world scenarios. Therefore, in this paper, we relax the requirement for the source-domain label spaces and propose a generalized version of MSMA named union-set multi-source model adaptation (US-MSMA). Specifically, in our US-MSMA setting, the union of all the source-domain label spaces instead of the label space of each source domain is required to be equal to the target domain label space. In other words, the label space of each source domain is just expected to be a subset but not necessarily the same as that of the target domain. Such a relaxation considerably extends the applicability of MSMA. Moreover, it also allows selection of source domains from a larger candidate set which may improve the adaptation performance. For example, a high-performance model that is trained in a source domain with high-quality labels of part of the target domain classes, can be used in the training of US-MSMA and contributes to improving the adaptation performance for certain classes. The generalized multi-source setting is especially compatible with model adaptation due to the low cost for introducing pre-trained models.

For handling the problem of US-MSMA for semantic segmentation, we propose a two-stage method which consists of a model adaptation stage and a model integration stage. For the model adaptation stage, we propose a novel learning strategy, model-invariant feature learning. Specifically, to take full advantage of diverse characteristics of the source-domain pre-trained models, we train the source-domain models to produce target-domain features with similar distributions which are referred to as model-invariant features. The conception of the model-invariant feature learning is to reduce the domain biases and improve the generalization ability of the source-domain models by harmonizing the model characteristics derived from different source domains. Moreover, to obtain predictions in the target-domain label space, we introduce a classifier ensemble strategy which combines predictions of all the classifiers of the source-domain

models as the complete prediction. To integrate the adapted source-domain models, we further introduce the model integration stage which distills knowledge of the adapted models to train a final model. We validate the effectiveness of our method in various situations of the union-set multi-source setting by conducting extensive experiments.

This paper's contributions are summarized as follows.

- We propose the problem setting of US-MSMA, which relaxes the requirement for the source-domain label spaces in the general multi-source setting and is thus applicable to a wider range of practical scenarios.
- We propose a two-stage method to handle the problem of US-MSMA for semantic segmentation. In the first stage, we propose the novel model-invariant feature learning for better generalization in the target domain. And as the second stage, we introduce the model integration to train a final model which absorbs knowledge from the adapted source-domain models.
- We conduct experiments in extensive adaptation settings that use several source-domain sets with different label space settings. Experimental results demonstrate the superiority of our method to previous adaptation methods.

2 Related Works

2.1 UDA for Semantic Segmentation

UDA for semantic segmentation has been widely studied in recent years. Typical technologies used in the methods mainly include image-to-image translation [14,16,4,29,32], adversarial learning [34,25,26,7,20], and semi-supervised learning [39,37,5,27,3]. Image-to-image translation is used for reducing the visual domain gap by modifying some image characteristics and is mainly performed with a GAN [10]-based model [38]. Adversarial learning introduces a domain discriminator for recognizing the domain of intermediate features [34] or final outputs [25]. Training the segmentation network and the discriminator against each other can align the feature distributions of the domains. Semi-supervised learning technologies can be readily applied to UDA due to the similar problem setting, including pseudo-label learning [39,37], self-ensembling [5], and entropy minimization [27,3]. In this paper, we also use the pseudo-label learning as the baseline of our method and introduce the entropy minimization for further improving the adaptation performance.

2.2 Multi-source UDA

Multi-source UDA methods have been developed for image classification [13,21,35], semantic segmentation [36,11], and object detection [33]. The technologies used in single-source UDA play a dominant role also in multi-source UDA. Zhao et al. [35] introduce multiple domain discriminators on the top of the feature extractor and conduct the adversarial learning between the feature extractor and the discriminators via a gradient reversal layer. Peng et al. [21] align moments

of feature distributions of the source domains and the target domain to transfer the source-domain knowledge. Zhao et al. [36] align the domains at both the pixel level and the feature level with the image-to-image translation and the adversarial learning. He et al. [11] also perform the pixel-level adaptation and further train with target-domain pseudo labels. Yao et al. [33] train a domain-adaptive object detector with multiple source subnets and obtain a target subnet by weighting and combining the parameters of the source subnets. The above methods are all developed for the general multi-source setting which is a particular case of our union-set multi-source setting.

2.3 Model Adaptation

Previous studies on model adaptation mainly focus on image classification, using some technologies similar to those used in UDA, such as generative models [15], adversarial learning [30], information maximization [17], and class prototypes [31]. As to semantic segmentation, Liu et al. [18] generate fake source-domain samples with real source-domain distribution to transfer the source-domain knowledge and introduce a patch-level self-supervision module for target-domain pseudo labels. Fleuret et al. [8] reduce uncertainty of predictions from multiple classifiers suffering from random noise to enhance robustness of the learned feature representation. Stan et al. [24] learn a prototypical distribution to encode source-domain knowledge and align the distributions across domains with the learned distribution. Model adaptation in the multi-source setting is studied poorly, with only one work [1] for image classification to the best our knowledge. Ahmed et al. [1] learn an optimal combination of multiple source-domain models with trainable weights, whereas the inference time is increased by several times due to the model combination.

3 Proposed Method

3.1 Problem Setting of US-MSMA

First, we detail the US-MSMA problem setting of our method. Let $\{D_i^S\}_{i=1}^k$ denote k labeled source domains with class sets $\{\Phi_i^S\}_{i=1}^k$ and D^T the unlabeled target domain with a class set Φ^T. Different from the general multi-source setting in which each of $\{\Phi_i^S\}_{i=1}^k$ must be absolutely the same as Φ^T, in our US-MSMA setting, the union of $\{\Phi_i^S\}_{i=1}^k$ is assumed to be equal to Φ^T, i.e., $\Phi_1^S \bigcup \Phi_2^S \cdots \bigcup \Phi_k^S = \Phi^T$. Given access to unlabeled data $\{x_i^T\}_{i=1}^n$ of D^T and k models $\{M_i\}_{i=1}^k$ pre-trained with $\{D_i^S\}_{i=1}^k$ respectively, we aim to obtain a model that learns knowledge transferred from $\{D_i^S\}_{i=1}^k$ to D^T and consequently achieves reasonable performance in D^T.

3.2 Overview of The Proposed Method

Two-stage architecture Figure 1 shows an overview of the proposed method which consists of the following two stages: a model adaptation stage and a model

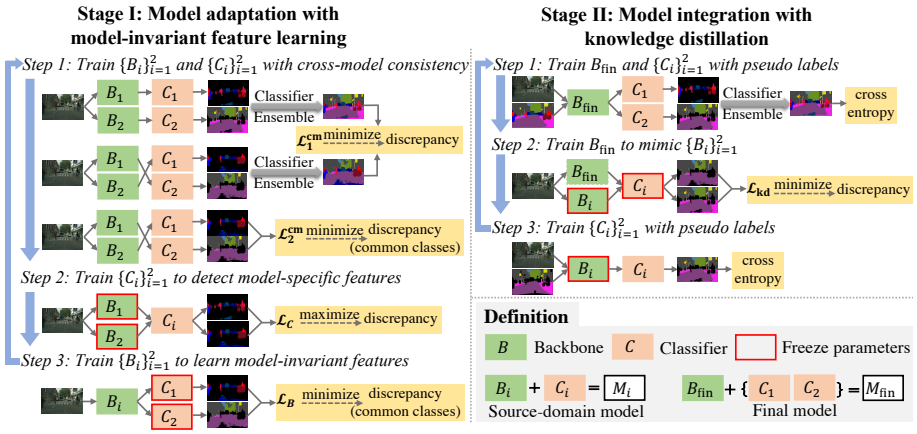

Fig. 1. An overview of the proposed method. For the ease of understanding, we show the case of using two source-domain models. The self-training with pseudo labels in Stage I is omitted in the figure.

integration stage. In Stage I, we conduct the model adaptation by retraining the pre-trained source-domain models $\{M_i\}_{i=1}^k$ with the target domain D^T. The source-domain knowledge is transferred to the target domain by training $\{M_i\}_{i=1}^k$ with pseudo labels of D^T in the manner of self-training (omitted in Fig. 1), and we improve the adaptation with the model-invariant feature learning. Then, in Stage II, we train a final model M_{fin} by distilling and integrating knowledge from $\{M_i\}_{i=1}^k$ trained in Stage I. As defined in Fig. 1, $\{M_i\}_{i=1}^k$ are individual models composed of a backbone B_i and a classifier C_i, and M_{fin} is an ensemble model composed of an integration backbone B_{fin} and all the classifiers $\{C_i\}_{i=1}^k$. The classifiers of the ensemble model are combined with a classifier ensemble strategy described below.

Classifier ensemble In both Stages I and II, we predict the probability distribution over all the target-domain classes Φ^T with the classifiers $\{C_i\}_{i=1}^k$. However, since the source-domain class sets $\{\Phi_i^S\}_{i=1}^k$ are not necessarily equal to Φ^T, the prediction over Φ^T may not be available from any individual C_i. And due to the possibly different class sets, the predictions of $\{C_i\}_{i=1}^k$ cannot be simply averaged. Therefore, to obtain the complete prediction, we use a classifier ensemble strategy which simultaneously combines and averages the predictions of $\{C_i\}_{i=1}^k$. Specifically, we calculate the unnormalized logits of the complete prediction by averaging the output logits of each class over $\{C_i\}_{i=1}^k$ as the following equation:

$$l_c(\cdot) = \frac{1}{\sum_{i=1}^k \mathbb{1}(c \in \Phi_i^S)} \sum_{i=1}^k C_{i,c}(\cdot), \ \forall c \in \Phi^T, \tag{1}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function, and $C_{i,c}(\cdot)$ denotes the logits of class c predicted by C_i if $c \in \Phi_i^S$ and is regarded as zero otherwise. The calculated logits are then normalized with the Softmax function as the predicted probability distribution in the target-domain label space. In such a manner, we obtain the prediction in the target-domain label space, using the classifiers $\{C_i\}_{i=1}^k$ with incomplete class sets instead of training a new classifier. The classifier ensemble operation is hereinafter denoted by $\mathrm{En}(\cdot)$.

3.3 Stage I: Model Adaptation with Model-invariant Feature Learning

As mentioned above, we conduct the model adaptation in Stage I on the basis of the self-training with the pseudo labels and introduce the model-invariant feature learning to improve the adaptation performance. The model-invariant feature learning aims to reduce the domain biases of $\{M_i\}_{i=1}^k$. Since the pre-trained $\{M_i\}_{i=1}^k$ initially have diverse characteristics derived from the source domains, the domain bias of each M_i can be reduced by harmonizing the characteristics of $\{M_i\}_{i=1}^k$. We realize it by training the backbones $\{B_i\}_{i=1}^k$ to produce features with similar distributions which we refer to as model-invariant features. The model-invariant features are learned by a iterative process composed of three steps: the first step for cross-model consistency and the subsequent two steps for adversarial learning between the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$, as shown in the left side of Fig. 1. We detail each component of Stage I as follows.

Self-training with pseudo labels To transfer the source-domain knowledge to the target domain, we perform the self-training for each of the individual models $\{M_i\}_{i=1}^k$ and also ensemble models that are composed of a backbone B_i $(i = 1, \ldots, k)$ and all the classifiers $\{C_i\}_{i=1}^k$. To this end, we use the pre-trained $\{M_i\}_{i=1}^k$ to generate pseudo labels $\{y_i\}_{i=1}^k$ in the source-domain label spaces and pseudo labels y^T in the target-domain label space for each image of D^T. To generate the pseudo labels in the target-domain label space, we combine and average the predictions of $\{M_i\}_{i=1}^k$ but not like the classifier ensemble described above since $\{M_i\}_{i=1}^k$ are pre-trained independently. Specifically, we first cast the probability distributions predicted by $\{M_i\}_{i=1}^k$ over the target-domain label space as $p_{i,c}(\cdot) = M_{i,c}(\cdot)$ if $c \in \Phi_i^S$ and $p_{i,c}(\cdot) = \frac{M_{i,0}(\cdot)}{\sum_{c' \in \Phi^T} \mathbb{1}(c' \notin \Phi_i^S)}$ otherwise, where $M_{i,c}(\cdot)$ is the probability of class c predicted by M_i and $M_{i,0}(\cdot)$ is the probability of the other classes not in Φ_i^S. Then we average the probability distributions $\{p_i\}_{i=1}^k$ and assign pseudo labels according to the average prediction. In the self-training, we use the cross-entropy loss function $\mathrm{CE}(logits, target)$ to train $\{M_i\}_{i=1}^k$ as follows:

$$\mathcal{L}_{\mathrm{pl}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} [\mathrm{CE}(C_i(B_i(x^T)), y_i) + \mathrm{CE}(\mathrm{En}(\{C_j(B_i(x^T))\}_{j=1}^k), y_T)], \quad (2)$$

where $\mathrm{En}(\cdot)$ is the classifier ensemble operation described in Section 3.2.

Cross-model consistency On the basis of the assumption that a backbone that produces model-invariant features should be compatible with any classifier, we randomly recombine $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ as k new models $\{C_{\mathrm{ma}(i)}(B_i(\cdot))\}_{i=1}^k$ where $\mathrm{ma}(i)$ is the index of the classifier matched with B_i. The recombined models are trained in terms of cross-model consistency which includes overall consistency of the ensemble predictions and per-class consistency of the logits output by individual classifiers. For the overall consistency, we perform the classifier ensemble operation for predictions of the original models $\{C_i(B_i(\cdot))\}_{i=1}^k$ and the recombined models $\{C_{\mathrm{ma}(i)}(B_i(\cdot))\}_{i=1}^k$ and minimize the discrepancy between the ensemble predictions with the following loss function:

$$\mathcal{L}_1^{\mathrm{cm}} = \mathbb{E}_{x^T \in D^T} ||\sigma(\mathrm{En}(\{C_i(B_i(x^T))\}_{i=1}^k)) - \sigma(\mathrm{En}(\{C_{\mathrm{ma}(i)}(B_i(x^T))\}_{i=1}^k))||_1, \quad (3)$$

where $\sigma(\cdot)$ is the Softmax function. For the per-class consistency, we train the recombined models to output consistent logits of each class. We calculate the average logits of each class and minimize the discrepancy between the output logits and the average logits as follows:

$$\mathcal{L}_2^{\mathrm{cm}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{c}^{\Phi_i^S} ||C_{\mathrm{ma}(i),c}(B_i(x^T)) - \delta_c(x^T)||_1, \quad (4)$$

where $\delta_c(\cdot)$ is the average logits of class c and calculated with the following equation:

$$\delta_c(\cdot) = \frac{1}{\sum_{i=1}^k \mathbb{1}(c \in \Phi_i^S)} \sum_{i=1}^{k} C_{\mathrm{ma}(i),c}(B_i(\cdot)), \quad (5)$$

where $C_{\mathrm{ma}(i),c}(\cdot)$ is the logits of class c output by $C_{\mathrm{ma}(i)}$ if $c \in \Phi_{\mathrm{ma}(i)}^S$ and zero otherwise. By recombining and training the models with $\mathcal{L}_1^{\mathrm{cm}}$ and $\mathcal{L}_2^{\mathrm{cm}}$ for the overall and the per-class consistency respectively, the features produced by the backbones are constrained to have similar distributions, i.e., to be model-invariant.

Adversarial learning In addition to the cross-model consistency, we further introduce adversarial learning between the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$ to enhance the model-invariant feature learning. Specifically, the adversarial learning consists of two steps: training $\{C_i\}_{i=1}^k$ to detect model-specific features and training $\{B_i\}_{i=1}^k$ to produce model-invariant features. The two steps are performed iteratively, and the parameters of $\{B_i\}_{i=1}^k$ ($\{C_i\}_{i=1}^k$) are frozen while updating those of $\{C_i\}_{i=1}^k$ ($\{B_i\}_{i=1}^k$).

For the training of C_i ($i = 1, \ldots, k$), we feed the features from B_i and B_j ($j \neq i$) into C_i and maximize the discrepancy between the predictions by minimizing the following loss function:

$$\mathcal{L}_C = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} [\sum_{j=1}^{k} -||C_i(B_i(x^T)) - C_i(B_j(x^T))||_1 + \mathrm{CE}(C_i(B_i(x^T)), y_i)], \quad (6)$$

where we add a cross-entropy term to prevent the recognition ability of $\{C_i\}_{i=1}^k$ from degradation while maximizing the discrepancy. The features from B_j that induce inconsistent predictions with those using the features from B_i, are considered domain-specific and detected by updating C_i with $\mathcal{L}_C$.

We train $\{B_i\}_{i=1}^k$ with a loss that calculates the discrepancy between the output logits and the average logits with features from each of the backbones respectively, as defined in the following equation:

$$\mathcal{L}_B = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{j=1}^{k} [\sum_{c}^{\Phi_j^S} ||C_{j,c}(B_i(x^T)) - \delta_{i,c}(x^T)||_1 + \mathrm{CE}(C_j(B_i(x^T)), y_j)], \quad (7)$$

where $\delta_{i,c}(\cdot)$ is the average logits of class c using the features from B_i and calculated as follows:

$$\delta_{i,c}(\cdot) = \frac{1}{\sum_{j=1}^{k} \mathbb{1}(c \in \Phi_j^S)} \sum_{j=1}^{k} C_{j,c}(B_i(\cdot)). \quad (8)$$

By minimizing the L1-norm term of $\mathcal{L}_B$, each of the backbones is trained to produce features that induce per-class consistent logits from different classifiers and are thus considered domain-invariant. However, since the L1-norm term involves only the classes shared by multiple classifiers, we additionally add a cross-entropy term to train each backbone to be compatible with all the classifiers.

Unlike the typical adversarial learning, we conduct the adversarial learning between two groups of the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$ rather than two specific opponents. Moreover, we train $\{B_i\}_{i=1}^k$ with a loss dissimilar to that for $\{C_i\}_{i=1}^k$, instead of minimizing the term $||C_i(B_i(x^T)) - C_i(B_j(x^T))||_1$ of $\mathcal{L}_C$ which may lead to excessive similarity among $\{B_i\}_{i=1}^k$. The adversarial learning is compatible with the cross-model consistency and can further enhance the model-invariance of the learned features.

With the components described above, the training of Stage I is performed by repeating a process of three steps. Step 1 simultaneously updates $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ with the loss $\mathcal{L}_{\mathrm{pl}}$ of the self-training and the losses $\mathcal{L}_1^{\mathrm{cm}}$ and $\mathcal{L}_2^{\mathrm{cm}}$ of the cross-model consistency. Then Step 2 and Step 3 updates $\{C_i\}_{i=1}^k$ and $\{B_i\}_{i=1}^k$ with $\mathcal{L}_C$ and $\mathcal{L}_B$, respectively.

3.4 Stage II: Model Integration with Knowledge Distillation

The models adapted in Stage I have slightly different performances, and it is generally unknown which one would perform best in the test. To achieve the best performance not inferring with all the models, we introduce the model integration stage that distills the knowledge of $\{M_i\}_{i=1}^k$ and trains a final model M_{fin}. M_{fin} is composed of an integration backbone B_{fin} and all the classifiers $\{C_i\}_{i=1}^k$, and the ensemble prediction $\mathrm{En}(\{C_i(B_{\mathrm{fin}}(\cdot))\}_{i=1}^k)$ is regarded as the final prediction. The parameters of the backbones $\{B_i\}_{i=1}^k$ are frozen in this stage.

Similar to Stage I, M_{fin} is trained with a loss that updates B_{fin} and $\{C_i\}_{i=1}^k$ together and losses that update B_{fin} and $\{C_i\}_{i=1}^k$ respectively. Specifically, B_{fin} and $\{C_i\}_{i=1}^k$ are trained together by minimizing $\text{CE}(\text{En}(\{C_i(B_{\text{fin}}(\cdot))\}_{i=1}^k), y_T)$, the cross-entropy loss for the ensemble prediction using pseudo labels that are generated with $\{M_i\}_{i=1}^k$ trained in Stage I. Then, B_{fin} is trained to mimic $\{B_i\}_{i=1}^k$ with a knowledge distillation loss defined as follows:

$$\mathcal{L}_{\text{kd}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \text{KLD}(C_i(B_{\text{fin}}(x^T)), C_i(B_i(x^T))), \tag{9}$$

where $\text{KLD}(input, target)$ is the Kullback–Leibler divergence function. Moreover, to maintain the compatibility of $\{B_i, C_i\}_{i=1}^k$, we train $\{C_i\}_{i=1}^k$ by minimizing $\sum_{i=1}^k \text{CE}(C_i(B_i(x^T)), y_i)$ for the predictions of individual classifiers. By training with the above losses, M_{fin} absorbs the knowledge of $\{M_i\}_{i=1}^k$ trained in Stage I and consequently outperforms all of $\{M_i\}_{i=1}^k$. Moreover, since $\{C_i\}_{i=1}^k$ are very light networks, the inference speed of M_{fin} is almost the same as that of a single model of $\{M_i\}_{i=1}^k$.

4 Experiments

4.1 Implementation Details

As the segmentation network in all the experiments, we used Deeplab V2 [2] with ResNet101 [12] of which the ResNet101 backbone and the atrous spatial pyramid pooling (ASPP) classifier were used as the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$, respectively. We trained the networks with a stochastic gradient descent (SGD) optimizer with an initial learning rate of 2.5×10^{-4} for 80,000 iterations. During the training, the learning rate was decreased with the poly policy with a power of 0.9, and the mini-batch size was set to 1. We used an equal weight of 1.0 for all the losses. Moreover, we also introduced a maximum squares loss [3] which minimizes the uncertainty of predictions, as a supplement to the losses $\mathcal{L}_{\text{pl}}$ and $\mathcal{L}_B$ in Stage I and also the loss $\text{CE}(\text{En}(\{C_i(B_{\text{fin}}(\cdot))\}_{i=1}^k), y_T)$ in Stage II to improve the adaptation performance. The code will be released.[1]

4.2 Datasets and Adaptation Settings

We used Synscapes dataset [28], GTA5 dataset [22], and Synthia dataset [23] as the source domains and Cityscapes dataset [6] as the target domain. All the source domains are synthetic datasets that are composed of photo-realistic images of street scenes, and Cityscapes is a real-world dataset of street-scene images. Synscapes, GTA5, and Cityscapes share a common label space containing 19 classes, while Synthia shares a subset containing 16 classes with the others. Synscapes, GTA5, and Synthia are hereinafter referred to as S, G, and T, respectively.

[1] `https://github.com/lzy7976/union-set-model-adaptation`

Table 1. Class distributions of the non-overlapping setting and the partly-overlapping setting. The target domain contains 19 classes: road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, bicycle.

Setting		Source domain	Background classes											Foreground classes							
			road	side.	buil.	wall	fence	pole	t-lig.	t-sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	moto.	bicy.
Non-overlapping	S+G	S												✓	✓	✓	✓	✓	✓	✓	✓
		G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓								
	$S(G)$+T	$S(G)$										✓		✓	✓	✓	✓	✓	✓	✓	✓
		T	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓								
	S+G+T	S														✓	✓	✓	✓		
		G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓								
		T												✓	✓					✓	✓
Partly-overlapping	S+G	S	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓		
		G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓
	$S(G)$+T	$S(G)$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓		
		T	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓					✓	✓

We conducted experiments with all the possible combinations of the source domains including S+G, S+T, G+T, and S+G+T. Furthermore, since our union-set multi-source setting is flexible in terms of the source-domain label spaces, we evaluated our method in various settings of the source-domain label spaces to demonstrate its effectiveness in a wide range of scenarios. Specifically, we conducted experiments in three label space settings: non-overlapping, partly-overlapping, and fully-overlapping. In the non-overlapping setting, the target-domain classes are divided into subsets with no common classes and assigned to the source domains. In the partly-overlapping setting, background classes are shared as common classes, while foreground classes are divided for the source domains. We detail the class distributions of the non-overlapping setting and the partly-overlapping setting in Table 1. In the fully-overlapping setting, all classes are shared expect three classes (terrain, truck, train) that are absent in Synthia, but we still involved these classes in the experiments of S+T and G+T.

4.3 Results in Non-overlapping Setting

Since no existing methods are proposed for such a problem setting, we slightly modified three methods proposed for related problem settings for comparisons: a UDA method [25] for semantic segmentation, a single-source model adaptation (SSMA) method [8] for semantic segmentation, and an MSMA method [1] for image classification. For the UDA method based on adversarial learning, we used domain-specific discriminators and classifiers for the source domains and trained one shared backbone. For both the UDA method and the SSMA method, the complete predictions were obtained by first casting the predictions in the source-domain label spaces over the target-domain label space and then averaging the

Table 2. Results in the non-overlapping setting. ST: self-training. CMC: cross-model consistency. ADV: adversarial learning. MSL: maximum squares loss. MI: model integration. PM: proposed method.

Method	S+G	S+T	G+T	S+G+T
ST	42.3	38.8	35.8	40.4
ST+CMC	43.2	39.1	36.1	40.6
ST+CMC+ADV	44.0	39.8	36.4	41.5
ST+CMC+ADV+MSL (**=Stage I of PM**)	45.8	41.4	37.2	42.1
ST+CMC+ADV+MSL+MI (**=PM**)	**46.6**	**42.3**	**37.9**	**44.2**
SSMA [8]	43.5	40.6	37.0	41.9
MSMA [1]	30.2	26.0	20.7	22.7
UDA [25]	45.7	39.2	35.9	41.1

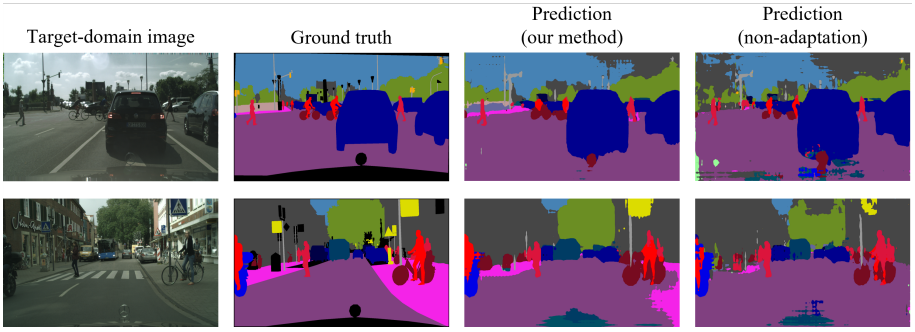

Fig. 2. Examples of the qualitative results of our method and the source-domain models without adaptation in the non-overlapping setting of S+G.

predictions, in the same manner as that for generating the pseudo labels. For the MSMA method which learns a set of weights for combining the models, we still cast the prediction of each model over the target-domain label space in the same manner and calculated the weighted average prediction with the learned weights. Moreover, for explicit comparisons with the SSMA method and the MSMA method, we performed the training of the methods using the same maximum squares loss and the cross-entropy loss with the same pseudo labels as those used in our method.

Table 2 shows the results in the non-overlapping setting. The mean Intersection over Union (IoU) over all the target-domain classes were used as the evaluation metric. For the incomplete versions of our method which trained multiple models but no final model, we independently evaluated the ensemble models composed of one backbone and multiple classifiers and reported the average performance. As shown in Table 2, the cross-model consistency and the adversarial learning, which are the two components of the model-invariant feature learning, successively improved the performance. By introducing the maximum squares loss, we obtained the results of Stage I of our method with further improvements. Finally, we performed the model integration of Stage II and achieved

Table 3. Results in the partly-overlapping setting. ST: self-training. CMC: cross-model consistency. ADV: adversarial learning. MSL: maximum squares loss. MI: model integration. PM: proposed method.

Method	*S*+*G*	*S*+*T*	*G*+*T*
ST	44.2	44.0	39.1
ST+CMC	46.0	44.8	39.6
ST+CMC+ADV	46.6	45.2	40.6
ST+CMC+ADV+MSL (**=Stage I of PM**)	47.4	45.9	42.2
ST+CMC+ADV+MSL+MI (**=PM**)	**48.3**	**47.2**	**43.5**
SSMA [8]	47.2	46.3	42.7
MSMA [1]	46.5	44.1	41.9
UDA [25]	47.9	46.9	41.6

Table 4. The per-class IoUs in the partly-overlapping setting for analyzing the influence of the model-invariant feature learning and the model integration on the performance over common and uncommon classes. ST: self-training. MIF: model-invariant feature learning, which is equal to ST+CMC+ADV in Table 3. S-I: Stage I of the proposed method. PM: proposed method.

Setting/ Method		IoUs of background classes (common)												IoUs of foreground classes (uncommon)								
		road	side.	buil.	wall	fence	pole	t-lig.	t-sign	vege.	terr.	sky	mean	pers.	rider	car	truck	bus	train	moto	bicy.	mean
S+*G*	ST	81.1	43.7	79.4	28.0	19.7	38.0	42.3	45.6	84.2	32.1	80.3	52.2	58.3	25.5	80.0	18.9	16.3	1.7	20.3	45.2	33.3
	MIF	86.4	46.2	82.1	34.6	27.4	37.7	42.5	43.4	85.0	40.2	84.2	55.4	58.8	26.8	78.4	23.6	20.9	2.2	21.0	44.0	34.5
	S-I	85.1	46.0	82.6	34.6	27.3	37.5	41.6	42.5	84.9	39.5	84.0	55.1	58.4	25.2	80.3	29.9	31.2	2.7	24.5	42.6	36.8
	PM	87.3	47.5	83.4	34.1	28.1	37.5	42.1	43.8	85.2	41.8	84.3	55.9	58.2	25.0	81.0	34.5	33.2	3.2	24.4	43.4	37.9
S+*T*	ST	73.8	36.2	82.1	26.3	19.4	36.8	37.2	38.7	84.4	27.6	82.7	49.6	60.8	22.3	82.1	22.2	18.3	1.9	27.5	55.9	36.4
	MIF	74.1	39.5	83.1	27.3	18.3	37.1	36.6	37.4	84.5	24.9	85.0	49.8	62.0	26.6	82.1	27.2	24.1	1.6	31.6	55.0	38.8
	S-I	76.8	39.4	83.5	26.6	20.8	37.3	38.0	37.3	85.0	21.3	83.7	50.0	62.7	31.3	81.7	27.3	35.0	1.4	27.0	55.9	40.3
	PM	80.6	42.5	84.1	27.2	23.4	37.6	37.0	39.6	85.2	23.2	84.6	51.4	62.4	31.9	82.2	31.8	37.1	1.9	30.4	54.0	41.5
G+*T*	ST	67.9	29.1	83.4	28.1	23.5	32.6	29.0	32.2	79.2	14.2	84.3	45.8	31.7	19.5	72.9	33.2	12.8	0.0	21.8	47.5	29.9
	MIF	73.9	32.3	83.5	28.8	22.3	33.2	26.8	26.5	78.6	15.9	86.4	46.2	35.1	20.6	82.0	36.1	16.3	0.0	24.2	48.8	32.9
	S-I	82.2	37.3	83.3	27.7	17.2	33.7	25.8	23.8	80.1	20.3	85.6	47.0	35.5	20.2	84.4	33.2	30.2	0.0	32.1	49.5	35.6
	PM	84.4	39.3	83.5	26.5	22.5	33.9	26.2	26.3	80.1	21.5	84.6	48.1	38.5	21.5	84.6	35.1	36.7	0.0	30.9	51.0	37.3

the best performance with significant improvements compared to the baseline trained with only the self-training. The above results for the ablation study validated the effectiveness of each component of our method.

Compared to other adaptation methods, our method outperformed all the others even with only Stage I in the non-overlapping setting. Moreover, both the SSMA and the MSMA methods need to infer with multiple models to obtain the complete predictions while the final model of our method has only one backbone and thus spends much less time for inference than the two methods. The MSMA method failed in the non-overlapping setting because the weighted model ensemble of the method is meaningless without any common classes.

Figure 2 shows two examples of the qualitative results of our method and the source-domain models without adaptation using source domains of *S*+*G*. It can be seen in the figure that the segmentation of both background and foreground classes was improved by our method. In the upper example, the predictions for sky, sidewalk, and vegetation became clearly better after the adaptation. And in the lower example, the classes including sidewalk, traffic sign, vegetation, and rider were segmented more precisely in the result of our method.

Table 5. Results in the fully-overlapping setting. ST: self-training. CMC: cross-model consistency. ADV: adversarial learning. MSL: maximum squares loss. MI: model integration. PM: proposed method.

Method	S+G	S+T	G+T
ST	47.6	45.7	44.0
ST+CMC	49.4	46.6	44.7
ST+CMC+ADV	50.9	47.7	45.7
ST+CMC+ADV+MSL (**=Stage I of PM**)	51.0	47.8	45.7
ST+CMC+ADV+MSL+MI (**=PM**)	**51.7**	48.7	**46.2**
SSMA [8]	49.8	47.6	45.5
MSMA [1]	**51.7**	47.2	44.2
UDA [25]	51.6	**49.0**	45.8

4.4 Results in Partly-overlapping Setting

For the partly-overlapping setting, we conducted the same experiments as those of the non-overlapping setting except S+G+T. The results are shown in Table 3. Similar to the results of the ablation study in the non-overlapping setting, each component contributed to the improvement consistently. Since both common and uncommon classes of the source domains exist in the partly-overlapping setting, we also show the per-class IoUs in Table 4 to analyze the influence of the model-invariant feature learning and the model integration on the performance over common and uncommon classes. By comparing "MIF" with "ST" in Table 4, the effectiveness of the model-invariant feature learning for both common and uncommon classes was indicated by the clear improvements except for the common classes of S+T. We think the slight improvement for the common classes of S+T was because S has a much less domain gap with the target domain for the background classes than that of T, and consequently harmonizing the model characteristics derived from S and T is not helpful to the generalization in the target domain. Moreover, the comparison between "S-I" and "PM" showed that the model integration of Stage II also improved the performance over both common and uncommon classes.

As to the comparisons to the other adaptation methods, our method achieved the best performance again as shown in Table 3. However, unlike the non-overlapping setting, the version with only Stage I of our method failed to outperform the SSMA and the UDA methods. Due to the presence of the common classes, the SSMA method benefited considerably from the model ensemble for obtaining complete predictions, which, however, decreased the inference speed by several times. Similarly, with the presence of the common classes, the MSMA method achieved reasonable performance with the weighted model ensemble. The UDA method achieved close performance to ours in S+G and S+T but required the access to the source-domain data.

4.5 Results in Fully-overlapping Setting

We conducted experiments in also the fully-overlapping setting which is exactly the general multi-source model adaptation. Due to the identical label space of the source domains and the target domain, we no longer performed model ensemble for the SSMA method and averaged the independent performances of all the trained models as the final performance. The MSMA method was evaluated with no changes since the weighted model ensemble acts as the core of the method. And for the UDA method, we used domain-specific discriminators but only one classifier. We did not conduct experiments in the S+G+T setting since compared to the performance in S+G, no improvements can be gained by introducing T due to the much larger domain gap between T and the target domain.

Table 5 shows the results in the fully-overlapping setting. Similarly to the other two settings, the results of the ablation study validated the effectiveness of each component of our method, with the only difference that the maximum squares loss made almost no contributions in the fully-overlapping setting. It can be explained by the fact that the pseudo labels generated in the fully-overlapping setting were more accurate than those in the other settings, which diminished the significance of the maximum squares loss.

In the comparisons to the other adaptation methods, our method still outperformed the SSMA method, while the UDA method achieved slightly better performance than ours in S+T. Moreover, the MSMA method achieved the same performance as ours in S+G using the two source domains with closer domain gaps with the target domain, which indicated that the efficiency of the weighted model ensemble is maximized with source domains that have the same label space and similar domain gaps with the target domain. Overall, our method has the best cost-performance ratio considering the inference speed and the requirement for the access to source-domain data.

5 Conclusion

This paper has presented a novel problem named union-set multi-source model adaptation, which requires the union of the source-domain label spaces to be equal to the target-domain label space and is thus applicable to a wider range of practical scenarios than that with the general multi-source setting. To tackle the problem of union-set multi-source model adaptation for semantic segmentation, we proposed a method with a novel learning strategy, model-invariant feature learning, to improve the generalization in the target domain by harmonizing the diverse characteristics of the source-domain models. Moreover, we further performed the model integration which distills the knowledge from the adapted models and trains a final model with improved performance. The effectiveness of each component of our method was validated by the results of the elaborate ablation studies, and the superiority of our method compared to previous adaptation methods was demonstrated by the results of the extensive experiments in various settings.

References

1. Ahmed, S.M., Raychaudhuri, D.S., Paul, S., Oymak, S., Roy-Chowdhury, A.K.: Unsupervised multi-source domain adaptation without access to source data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10103–10112 (2021)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2017)
3. Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2090–2099 (2019)
4. Chen, Y.C., Lin, Y.Y., Yang, M.H., Huang, J.B.: Crdoco: Pixel-level domain transfer with cross-domain consistency. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1791–1800 (2019)
5. Choi, J., Kim, T., Kim, C.: Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6830–6840 (2019)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016)
7. Du, L., Tan, J., Yang, H., Feng, J., Xue, X., Zheng, Q., Ye, X., Zhang, X.: Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 982–991 (2019)
8. Fleuret, F., et al.: Uncertainty reduction for model adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9613–9623 (2021)
9. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the International Conference on Machine Learning. pp. 1180–1189 (2015)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in Neural Information Processing Systems **27** (2014)
11. He, J., Jia, X., Chen, S., Liu, J.: Multi-source domain adaptation with collaborative learning for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11008–11017 (2021)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
13. Hoffman, J., Mohri, M., Zhang, N.: Algorithms and theory for multiple-source adaptation. Advances in Neural Information Processing Systems **31** (2018)
14. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: Proceedings of the International Conference on Machine Learning. pp. 1989–1998 (2018)
15. Li, R., Jiao, Q., Cao, W., Wong, H.S., Wu, S.: Model adaptation: Unsupervised domain adaptation without source data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9641–9650 (2020)

16. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6936–6945 (2019)
17. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the International Conference on Machine Learning. pp. 6028–6039 (2020)
18. Liu, Y., Zhang, W., Wang, J.: Source-free domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1215–1224 (2021)
19. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: Proceedings of the International Conference on Machine Learning. pp. 97–105 (2015)
20. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2507–2516 (2019)
21. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1406–1415 (2019)
22. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Proceedings of the European Conference on Computer Vision. pp. 102–118 (2016)
23. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3234–3243 (2016)
24. Stan, S., Rostami, M.: Unsupervised model adaptation for continual semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2593–2601 (2021)
25. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7472–7481 (2018)
26. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1456–1465 (2019)
27. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2517–2526 (2019)
28. Wrenninge, M., Unger, J.: Synscapes: A photorealistic synthetic dataset for street scene parsing. arXiv preprint arXiv:1810.08705 (2018)
29. Wu, Z., Han, X., Lin, Y.L., Uzunbas, M.G., Goldstein, T., Lim, S.N., Davis, L.S.: Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 518–534 (2018)
30. Xia, H., Zhao, H., Ding, Z.: Adaptive adversarial network for source-free domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9010–9019 (2021)

31. Yang, S., Wang, Y., van de Weijer, J., Herranz, L., Jui, S.: Unsupervised domain adaptation without source data by casting a bait. arXiv preprint arXiv:2010.12427 (2020)
32. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4085–4095 (2020)
33. Yao, X., Zhao, S., Xu, P., Yang, J.: Multi-source domain adaptation for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3273–3282 (2021)
34. Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Fully convolutional adaptation networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6810–6818 (2018)
35. Zhao, H., Zhang, S., Wu, G., Moura, J.M., Costeira, J.P., Gordon, G.J.: Adversarial multiple source domain adaptation. Advances in Neural Information Processing Systems **31**, 8559–8570 (2018)
36. Zhao, S., Li, B., Yue, X., Gu, Y., Xu, P., Hu, R., Chai, H., Keutzer, K.: Multi-source domain adaptation for semantic segmentation. Advances in Neural Information Processing Systems **32**, 7287–7300 (2019)
37. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. International Journal of Computer Vision **129**(4), 1106–1120 (2021)
38. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232 (2017)
39. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision. pp. 289–305 (2018)