博士論文

A Study on Practicality Improvement of Image Recognition Technologies by Mitigation of Label Dependence

ラベル依存緩和による画像認識技術の実用性向上に関する研究

北海道大学 大学院情報科学院
情報科学専攻
メディアネットワークコース

李 宗曜

2023年8月

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the past years, deep learning technologies have been developed rapidly and shown increasing application potential for real-world problems. In the field of computer vision, deep learning methods have made remarkable achievements, especially for the basic image recognition tasks such as image classification [1], object detection [2], and semantic segmentation [3]. However, there is still an obstacle to the applications of deep learning to a wide range of real-world problems, label dependence for training the deep learning models. Typically, the deep learning models with outstanding performance are trained on the basis of supervised learning, thereby heavily relying on a large amount of well-labeled data that are unavailable in many real-world problems. Medical image analysis is a typical field facing such a problem, where a large-scale well-labeled dataset of medical images is always hard to collect because it requires specialized knowledge to annotate the medical images and it may be difficult to share the medical images due to privacy concerns. Moreover, for the image recognition tasks of which the aim is not simply classifying an image, such as object detection and semantic segmentation, it may take a long time to precisely annotate an image. For example, in the Cityscapes dataset [4], which consists of images of urban street scenes with fine pixel-level annotations, it requires more than one and a half hours on average to annotate a single image. Constructing a large-scale dataset for such tasks is expensive and can considerably improve

the costs of applying the deep learning technologies. Therefore, it is necessary to mitigate the label dependence for applying the deep learning technologies to a wide range of real-world problems.

Studies for mitigating the label dependence have been done in various directions, such as unsupervised learning, semi-supervised learning, weakly-supervised learning, unsupervised domain adaptation, and model adaptation. The similarity of most of the studies is the use of unlabeled data, while the rests of the problem settings are different. Unsupervised learning tries to discover data structures and hidden patterns from totally unlabeled data for clustering or analyzing the unlabeled data. Using only unlabeled data is ideal for getting rid of the label dependence but can hardly be applied to real-world problems due to the high difficulty of the extreme problem setting. Semi-supervised learning uses a dataset comprised of both labeled and unlabeled data and aims to train a model with performance as close as possible to the model trained with the fully-labeled dataset. As a result, the need for labeled data is reduced. Weakly-supervised learning mitigates the label dependence by using weak supervisions to train the models rather than using totally unlabeled data. In some literature, semi-supervised learning is also categorized as one of the weakly-supervised learning problems, but here the weak supervisions are used to refer to coarse-grained annotations, such as image-level labels for object detection or semantic segmentation. By using the coarse labels, the annotation costs of constructing the datasets can be greatly reduced. Unsupervised domain adaptation is similar to semi-supervised learning in respect of using both labeled and unlabeled data but introduces domain shifts between the labeled and unlabeled data. The domain shifts denote the differences in distribution between the training and test data that are assumed to be consistently distributed in general machine learning problems. Unsupervised domain adaptation aims to transfer the label knowledge from the labeled data (referred to as source domain) to the unlabeled data (referred to as target domain), and therefore in some applications, it is possible to use labeled data of a different domain as an alternative to constructing a labeled dataset of the target domain. Model adaptation (also called source-data-free domain adaptation) is a more challenging variant of unsupervised domain adaptation. Rather than transferring the

knowledge from the source-domain data, model adaptation transfers the knowledge from a pre-trained source-domain model without using the source-domain data. Model adaptation is more practical than unsupervised domain adaptation because access to the pre-trained models is always easy to obtain while the source-domain data may be inaccessible due to privacy policy or storage limitation.

1.2 Proposition in this Thesis

This thesis focuses on three of the above-mentioned study directions, semi-supervised learning, unsupervised domain adaptation, and model adaptation, which are promising for mitigating the label dependence in the real-world applications of deep learning. For semi-supervised learning, because medical image analysis is one of the most promising application fields of semi-supervised learning, this thesis proposes a pseudo-label-based semi-supervised learning method to solve a problem of medical image analysis, chronic gastritis classification using gastric X-ray images. For unsupervised domain adaptation, this thesis proposes several methods for two important computer vision tasks, object detection and semantic segmentation. For the model adaptation, this thesis proposes the first solution to multi-source model adaptation of semantic segmentation, where multiple pre-trained models of different source domains are used for model adaptation. This thesis studies on model adaptation in the multi-source setting because when more than one source domains are available it may be difficult to choose the optimal one without accessing to the source-domain data, and using multiple source-domain models is a natural solution. Moreover, this thesis solves a problem of the multi-source setting, the strict requirement for label spaces. The previous multi-source setting requires the label spaces of all the source domains to be equal to that of the target domain. However, this requirement is so strict that the multi-source methods can be hardly applied to real-world problems. To improve the practicality, this thesis proposes a novel multi-source setting where the requirement for the label spaces are relaxed. Specifically, in the relaxed multi-source setting, the union set of the source-domain label spaces is required to be equal to the target-domain label space, while the label spaces of the single source domains can be different subsets of the

target-domain label space. For the new multi-source setting, this thesis also proposes the first model adaptation method.

The methods proposed in this thesis are summarized as follows:

1. For chronic gastritis classification using gastric X-ray images, this thesis proposes a semi-supervised learning method based on tri-training. Tri-training [5] is a pseudo-label learning technique that improves the self-training by using multiple models to improve the reliability of the pseudo labels. Moreover, Between-Class learning [6], a data augmentation technique, is introduced for enhancement of the semi-supervised learning performance.

2. For unsupervised domain adaptation of object detection, this thesis proposes a method based on the adversarial learning which aligns the feature distributions of the source domain and the target domain to reduce the domain shift. Compared to the previous adversarial learning-based method [7], the adversarial learning in the proposed method is guided to focus on foreground regions and poorly-aligned regions, thereby improving the feature alignment performance.

3. For unsupervised domain adaptation of semantic segmentation, this thesis proposes a method that performs symmetric adaptation with the adversarial learning. The symmetric adaptation is to train two models with the adversarial learning using two symmetric adversarial losses. Moreover, the proposed method uses the prediction consistency of the two models to improve the reliability of the pseudo labels.

4. For unsupervised domain adaptation of semantic segmentation, this thesis proposes a second method that uses a variational autoencoder [8] as a replacement of the adversarial learning. The variational autoencoder is used to learn the output distribution of the segmentation model and align the distributions between the source and target domains.

5. For unsupervised domain adaptation of semantic segmentation, this thesis proposes a third method that learns intra-domain style-invariant representation. The proposed method first trains a novel semantic-aware multimodal image-to-image translation model

for obtaining images with diverse intra-domain styles and then trains the segmentation model with a self-ensembling method based on consistency regularization.

6. For multi-source model adaptation of semantic segmentation, this thesis proposes a method that aims to learn model-invariant features, i.e., features with similar distributions from the pre-trained source-domain models. Once the feature distributions of different source-domain models are aligned, the target-domain model harmonizes the characteristics of the source domains and is thus more generalizable to the target domain.

7. For model adaptation of semantic segmentation in the new multi-source setting where the requirement for the label spaces is relaxed, this thesis proposes a method that is constructed on the basis of the method 6 for the general multi-source setting. The method 6 is modified to adapt to the new multi-source setting, and the conception of learning the model-invariant features remains unchanged.

1.3 Organization of this Thesis

This thesis is comprised of six chapters. The rest of this thesis is organized as follows. In Chapter 2, related works of semi-supervised learning, unsupervised domain adaptation, and model adaptation are presented, and the most representative ones are listed. Chapters 3, 4 and 5 present the methods for semi-supervised learning, unsupervised domain adaptation, and model adaptation, respectively. In chapter 3, the semi-supervised learning method 1 for chronic gastritis classification using gastric X-ray images is presented. Chapter 4 presents the method 2 for unsupervised domain adaptation of object detection and the methods 3, 4 and 5 for unsupervised domain adaptation of semantic segmentation. Chapter 5 presents the methods 6 and 7 for multi-source model adaptation of semantic segmentation. For each method presented in Chapters 3, 4 and 5, complete experiments are conducted for validating the effectiveness of the proposed methods. Finally, Chapter 6 makes a conclusion of this thesis and discusses the future directions.

Chapter 2

Related Works

2.1 Introduction

This chapter presents related works of this thesis. Focusing on mitigating the label dependence, this thesis presents studies on semi-supervised learning, unsupervised domain adaptation, and model adaptation which is a variant of unsupervised domain adaptation. Therefore, this chapter presents the previous works in the above three areas. Specifically, Section 2.2 presents the previous works of semi-supervised learning. Because the ideas of many semi-supervised learning methods for natural images are also effective for medical images, and methods for medical images may be customized according to the specific task, Section 2.2 presents only the semi-supervised learning methods for natural images. Section 2.3 presents the previous works of unsupervised domain adaptation. Since the proposed methods of this thesis for unsupervised domain adaptation tackle the tasks of object detection and semantic segmentation, the previous methods for the two tasks are presented in Section 2.3.1 and Section 2.3.2, respectively. Section 2.4 presents the previous works of model adaptation, mainly focusing on those for semantic segmentation. Since model adaptation has been rarely studied in the multi-source setting, Section 2.4 presents only the model adaptation methods for the single-source setting.

2.2 Semi-supervised Learning

Semi-supervised learning methods have been developed using various technologies, such as generative models [9–11], entropy minimization [12, 13], and consistency regularization [14–19]. Generative models are used for generating new samples that the task model can not recognize well, and the recognition performance of the model trained with the generated samples is consequently improved. Entropy minimization tries to minimize the conditional entropy of the model predictions so that the model becomes more confident and more accurate. Pseudo-label learning can be regarded as a form of entropy minimization because the predictions are enforced to be confident by using loss functions such as cross entropy. Consistency regularization based on the smoothness assumption is the state-of-the-art technology of semi-supervised learning. The regularization is typically on the basis of perturbations to the unlabeled samples, while some methods also use the Mixup [20] augmentation technique. Due to the superior performance of the consistency regularization-based methods, some representative ones of the methods are listed and described as follows.

Reference [14]

The method proposed in the reference [14] uses normal perturbations such as Gaussian noise to produce augmented copies and computes the differences between the predictions of two augmented copies of the same original sample as the consistency regularization loss. Specifically, the prediction of one of the augmented copies is the current model's prediction, and that of the other one can be either the current model's prediction or the temporal ensemble prediction which is a moving average of the history predictions.

Reference [16]

The method proposed in the reference [16] replaces the normal perturbations in the reference [14] with an adversarial learning-based perturbation. Specifically, the perturbation is derived with the adversarial learning to make the prediction on the augmented copy to most greatly deviate from the correct label. With the adversarial learning-based

perturbation, the model's robustness is enhanced; hence the model can perform better on unseen samples.

Reference [19]

The method proposed in the reference [19] uses the Mixup [20] augmentation technique to produce interpolation samples each of which is an interpolation of two different unlabeled samples. The model makes predictions for the interpolation samples and the original samples, and the consistency regularization loss is computed between the prediction for the interpolation sample and the interpolation of the predictions for the two original samples. The effectiveness of the interpolation-based consistency regularization is even better than that of the perturbation-based one with the adversarial learning.

2.3 Unsupervised Domain Adaptation

Unsupervised domain adaptation methods have been extensively developed for image classification, of which the representative methods typically align the feature distributions of the source domain and the target domain, measuring the distribution distance with the maximum mean discrepancy [21] and the adversarial learning [22]. For object detection and semantic segmentation, prevalent technologies are similar including adversarial learning, image-to-image translation, and semi-supervised learning. The adversarial learning is used for the feature alignment, which trains a discriminator to recognize the domain of the features and trains the task model to fool the discriminator. The image-to-image translation transfers the image styles across domains to reduce the domain shift at the image level. Some semi-supervised learning technologies can be also applied to unsupervised domain adaptation due to the similar problem setting. In the rest of this section, the previous works on unsupervised domain adaptation of object detection and semantic segmentation are presented respectively.

2.3.1 Unsupervised Domain Adaptation of Object Detection

Previous methods on unsupervised domain adaptation of object detection are mainly proposed for the prevailing two-stage object detector, Faster R-CNN [2]. Adversarial learning-based methods [7,23–26] align the global-level and instance-level features. The previous methods [27, 28] use the unpaired image-to-image translation model CycleGAN [29] to generate synthetic source-like target-domain images (or target-like source-domain images) for bridging the domain gap. The semi-supervised learning technologies applied to the cross-domain object detection include pseudo-label learning [30, 31] and Mean-Teacher-based consistency regularization [32, 33]. Some methods [34, 35] compute class-wise prototypes to estimate the source and target distributions. Moreover, some methods [36, 37] introduce an auxiliary classifier to explore the categorical information. Three representative methods are described as follows.

Reference [7]

The method proposed in the reference [7] attaches a gradient reversal layer before the global-level and instance-level discriminators into which the backbone features and the instance features are fed. The gradients are reversed during the backpropagation to perform the adversarial learning at the global level and the instance level.

Reference [31]

The method proposed in the reference [31] introduces a collaborative training manner on the basis of the adversarial learning. Specifically, the object detector consists of a backbone network, a region proposal network (RPN) and a region proposal classifier (RPC). The method uses the predictions of the RPC to produce pseudo labels for the RPN and uses the predictions of the RPN to weight the entropy minimization loss of the RPC. Moreover, the RPN and RPC are trained to maximize the prediction discrepancy between the RPN and RPC while the backbone is trained to minimize the discrepancy for better aligning the feature distributions.

Reference [35]

The method proposed in the reference [35] produces class-wise prototype representations with the region proposals through graph-based information propagation. Then, the model is trained with a contrastive loss that brings the prototypes of the same categories closer and pushes the prototypes of different categories further from each other.

2.3.2 Unsupervised Domain Adaptation of Semantic Segmentation

Previous methods on unsupervised domain adaptation of semantic segmentation have similar components to those for object detection, which include adversarial learning, image-to-image translation, and semi-supervised learning. The adversarial learning can be performed at either the intermediate feature level [38] or the segmentation output level [39]. For the image-to-image translation which reduces the visual differences between the source and target domains, CycleGAN [29] based on the generative adversarial network is most widely used, and other technologies include style transfer [40, 41] and Fourier Transform [42]. The most effective semi-supervised learning technology is the pseudo-label learning, which can be further improved with class-balanced pseudo labels [43], uncertainty-based rectification [44], and confidence-based regularization [45]. In addition, entropy minimization and Mean-Teacher-based consistency regularization are also shown to be effective in the methods [46, 47] and the method [48], respectively. Three representative methods are described as follows.

Reference [39]

The method proposed in the reference [39] performs the adversarial learning at the output level. Specifically, a discriminator is trained to recognize the domain of the segmentation outputs, and the segmentation model is trained to make the discriminator recognize the target-domain outputs as those of the source domain.

Reference [43]

The method proposed in the reference [43] tries to tackle the class imbalance problem of the pseudo-label learning. The method uses confidence (i.e., probability) thresholds to select reliable pseudo labels and improve the pseudo labels in terms of class balance

by using an independent threshold for each class to ensure that enough pseudo labels of the rare classes are selected.

Reference [49]

The method proposed in the reference [49] combines the adversarial learning, the pseudo-label learning, and the image-to-image translation. The segmentation model is introduced into the learning of the image-to-image translation model, and the learning of the two models are conducted alternatively to promote each other, which is the novelty of the method.

2.4 Modal Adaptation

To get rid of the dependence on the access to the source-domain data, model adaptation which only requires the pre-trained source-domain model is obtaining more and more attentions. The model adaptation methods have been developed for image classification so far, and the other tasks such as semantic segmentation are not sufficiently studied. The ideas for model adaptation of image classification include generating target-style data with generative models [50], using information maximization [51], aligning feature prototypes using the source classifier [52], and distinguishing source-similar/dissimilar features with adversarial learning [53]. Recent methods also estimate the source-domain feature distribution with target-domain anchors [54], explore domain-invariant parameters of the source-domain model [55], and use knowledge distillation with Mixup-based regularization [56]. Unfortunately, for semantic segmentation, the methods for image classification are less applicable and less effective due to the greater difficulty of pixel-level classification.

For model adaptation of semantic segmentation, some methods try to estimate the unseen source-domain distribution by using a Gaussian mixture model in the embedding space [57] and training a generative model with the source-domain feature statistics [58]. Other ideas for improving the model adaptation performance include uncertainty and prior distribution-aware intra-domain adaptation [59], historical contrastive learning [60], and reducing prediction un-

certainty of multiple classifiers [61]. Three of the methods are described as follows.

Reference [57]

The method proposed in the reference [57] estimates a prototypical distribution while training the source-domain model and aligns the target-domain distribution with the source-domain distribution in the embedding space using the estimated prototypical distribution while training the target-domain model. The prototypical distribution is modeled as a Gaussian mixture model, and the source and target distributions are aligned in the embedding space by minimizing the Sliced Wasserstein Distance.

Reference [61]

The method proposed in the reference [61] enhances the robustness of the backbone network by attaching multiple auxiliary classifiers into which backbone features corrupted by dropout are fed and reducing the uncertainty between the main and auxiliary predictions. By enforcing the predictions of the auxiliary classifiers to be consistent with those of the main classifier, the noise robustness of the feature representation is enhanced, thereby improving the generalization.

Reference [60]

The method proposed in the reference [60] uses contrastive learning to learn instance-discriminative target representations. The method obtains the features of the query sample with the current model and features of the key samples with a historical model. Like other contrastive learning methods, a contrastive loss is used to pull the query close to the positive keys while pushing it far from the negative keys.

2.5 Problems to Be Solved in this Thesis

This section clarifies the problems to be solved in this thesis. Since this thesis solves the problem of the label dependence in three directions including semi-supervised learning, unsupervised domain adaptation, and model adaptation, the specific problems to be solved in the three directions are described respectively as follows.

Previous studies on semi-supervised learning are mainly conducted with natural images while semi-supervised learning for medical images is not studied sufficiently. It is expected to validate whether the semi-supervised learning technologies are effective for the medical images. For chronic gastritis classification using gastric X-ray images, which is chosen as the task in this thesis, the state-of-the-art technology, perturbation-based consistency regularization, can be hardly applied because the gastric X-ray images are very fine-grained and the perturbations such as Gaussian noise may corrupt the key details for the gastritis classification. Therefore, pseudo-label learning seems to be the most appropriate way for the task. However, the traditional pseudo-label learning that produces the pseudo labels with the model itself faces a limitation. The model's performance can be hardly improved by producing new pseudo labels that are more accurate than the old ones with the newly trained model because the model may become more confident in the incorrect predictions using the pseudo labels produced by itself. Such a limitation is a huge obstacle to achieving progressive performance improvements, and this thesis tries to solve the problem by introducing a tri-training mechanism, where three heterogeneous models are trained and the pseudo labels for each model are produced with the other two models.

For unsupervised domain adaptation, the problems to be solved in object detection and semantic segmentation are different. First, for unsupervised domain adaptation of object detection, the global feature alignment of the previous method is unaware of two important information: (1) whether the features are in background region or foreground region, and (2) whether the features are well aligned or poorly aligned. Without the information, the feature alignment pays equal attention to all the regions and is thus less effective. To solve this problem, this thesis introduces a divergence-based guidance mechanism to make the feature alignment to pay more attention to the foreground regions and poorly-aligned regions that are more significant. Second, for unsupervised domain adaptation of semantic segmentation, this thesis proposes three methods to solve three problems respectively: (1) noise in the pseudo labels, (2) difficulty of training with the adversarial learning, and (3) neglect of diversity within the target domain. For the first problem, this thesis proposes a method that performs symmetric

adversarial learning to produce prediction similarity maps that are involved in the thresholds to reduce the noise in the pseudo labels. As to the second problem, because the optimization of the adversarial learning is always difficult and unstable, this thesis proposes an alternative method based on variational autoencoder that can learn the feature distribution similarly and is easier to perform than the adversarial learning. Finally, for the third problem, because the previous methods do not consider the intra-domain diversity within the target domain, which is significant for the generalization of the trained model, this thesis proposes a method that learns intra-domain style-invariant features to improve the generalization in the target domain.

Almost all the studies on model adaptation are conducted in the single-source setting, while multi-source model adaptation which can be practical in real-world applications is remained to be studied. Therefore, this thesis proposes the first method for multi-source model adaptation of semantic segmentation. Moreover, as mentioned in Section 1.2, the previous multi-source setting is too strict to be applied to real-world problems. To make the methods for multi-source model adaptation applicable to a wider range of scenarios, this thesis relaxes the requirement for the label spaces and realizes model adaptation in the new multi-source setting.

2.6 Conclusion

This chapter has presented the related works of the thesis, including studies on semi-supervised learning, unsupervised domain adaptation, and model adaptation. Moreover, this chapter has clarified the problems to be solved in this thesis and briefly introduced the solutions.

Chapter 3

Mitigation of Label Dependence with Semi-supervised Learning

3.1 Introduction

This chapter proposes a semi-supervised learning method for mitigating the label dependence in the task of chronic gastritis classification using gastric X-ray images. Chronic gastritis is known as a factor that may cause gastric cancer, and clinicians can diagnose chronic gastritis and identify the risk of gastric cancer with gastric X-ray images [62]. To reduce the burden of reading many gastric X-ray images on clinicians, models that can automatically recognize gastritis from the X-ray images is needed. Although the previous work [63] has realized gastritis classification with a deep convolutional neural network (CNN), the method is on the basis of supervised learning and thus depends on a large number of gastric X-ray images being annotated by experts. For mitigating the label dependence of the gastritis classification with semi-supervised learning, as described in Section 2.5, the methods based on perturbation-based consistency regularization are less useful because the perturbations may corrupt the key details in the X-ray images. Therefore, we develop the method on the basis of the pseudo-label learning which is applicable to a wider range of scenarios. General pseudo-label learning trains the model with the pseudo labels produced with the model itself, and the performance can be hardly improved by performing the pseudo-label learning iteratively. To solve the problem, we introduce the tri-training architecture, where three heterogeneous

models are trained and the pseudo labels for each model are produced with the other two models. Moreover, we use the Between-Class (BC) learning [6], a data augmentation strategy, to enhance the semi-supervised learning performance.

3.2 Semi-supervised Learning Based on Tri-training for Chronic Gastritis Classification

To classify the X-ray images as gastritis or non-gastritis, we construct a patch-based method. Specifically, we crop the whole X-ray images into small patches and classify the patches into three classes: gastritis patch, non-gastritis patch, and irrelevant patch (i.e., regions outside the stomach). Then, we classify the whole X-ray image with a majority voting system that considers only the patches classified as gastritis or non-gastritis of the X-ray image. We develop our semi-supervised learning method on the basis of the tri-training architecture and equip the method with a data augmentation strategy known as BC learning. The procedures of our method are illustrated in Fig. 3.1.

3.2.1 Tri-training Architecture

Tri-training is a semi-supervised learning algorithm proposed by Zhou et al. [5] that aims to improve the performance of discriminative models. Originally designed for traditional machine learning models like SVM and random forest [64], it can also be applied to deep learning models. The core idea behind tri-training is to augment the labeled dataset by incorporating unlabeled data along with their predicted labels. The algorithm begins by training three models using separate initial training sets. These models serve as a starting point for the iterative process consisting of two steps. Firstly, each model's training set is augmented by adding unlabeled samples that are consistently classified as the same class by the other two models. These unlabeled samples are assigned consistent predicted labels. Secondly, the three models are retrained using their respective augmented training sets obtained in the previous step. Notably, tri-training introduces a unique augmentation architecture where the training set of

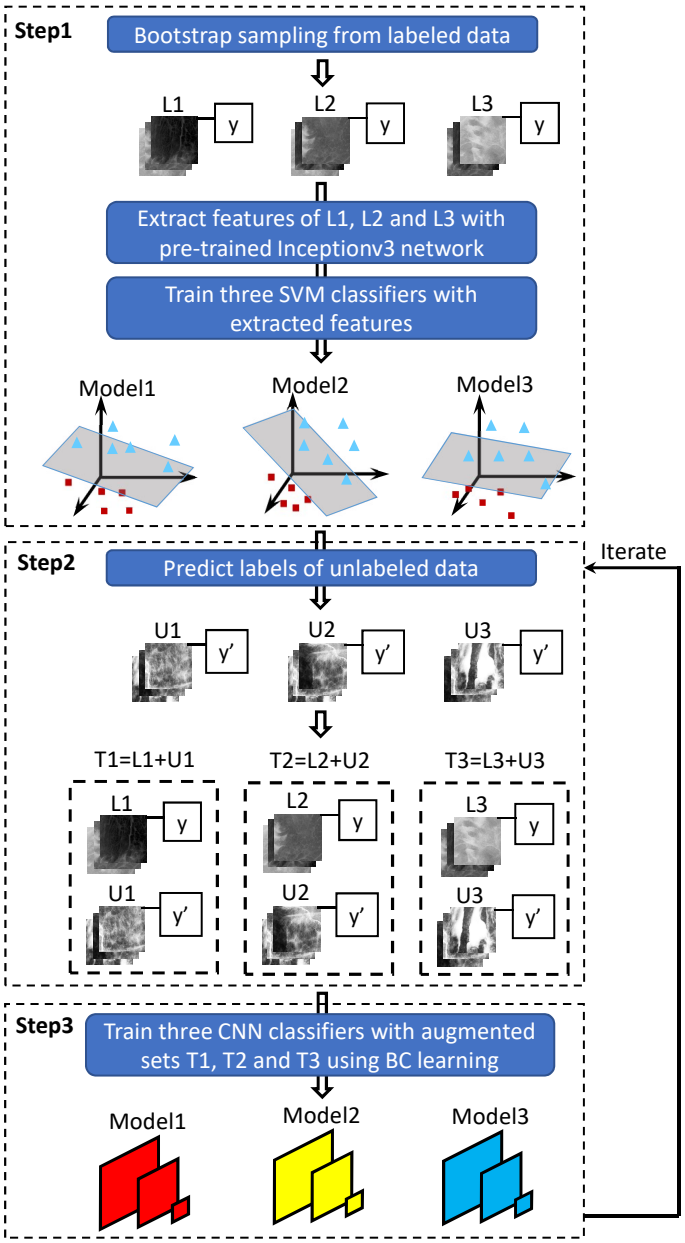


Figure 3.1: Procedures of the proposed method. 'L1', 'L2' and 'L3' are training sets randomly sampled from labeled data. 'U1', 'U2' and 'U3' are training sets with pseudo labels selected from unlabeled data. Specifically, 'U1' consists of samples that are predicted as the same class by 'Model2' and 'Model3', 'U2' consists of samples that are predicted as the same class by 'Model1' and 'Model3', and 'U3' consists of samples that are predicted as the same class by 'Model1' and 'Model2'. 'y' and 'y'' denote the ground truth label and the pseudo label, respectively.

each model is expanded based on the predictions of the other two models. This approach does not rely on setting a probability threshold to select unlabeled samples. Instead, the consistent predictions from two models are considered reliable enough for training, eliminating the need for redundant hyper-parameters. The tri-training algorithm can be visualized using Fig. 3.1 and can be summarized in the following three steps.

- **Step 1.** We start by creating three training sets using the bootstrap method [65]. These training sets are sampled from the labeled data. We then train three SVM classifiers using these training sets. To extract the features for training the SVMs, we use the Inceptionv3 network [66], which has been pre-trained on the Imagenet dataset [67].

- **Step 2.** Next, we use the three trained models (SVMs or CNNs) to predict the labels of unlabeled samples. If two models classify an unlabeled sample into the same class, we add that sample to the training set of the remaining model.

- **Step 3.** We now train three CNN classifiers using the augmented training sets obtained from Step 2. After training, we repeat Step 2, and this iterative process is performed for a certain number of times.

Note that we start with SVMs as the initial models due to the small size of the labeled dataset. CNNs trained with a limited number of samples often suffer from overfitting and instability. In Step 3, all the models trained with augmented training sets are CNNs. To increase diversity and robustness, we employ three different network structures: ResNet [1], DenseNet [68], and a basic architecture with convolutional and fully connected layers.

By augmenting the training sets with unlabeled data, even though there may be some noisy samples with incorrect labels, the models trained on these augmented sets consistently outperform models trained solely on labeled data, especially when labeled data is scarce. With these improved models, higher-quality augmented training sets are obtained, leading to further performance improvements. The number of iterations can be determined based on the similarity among the three models' predictions for the unlabeled data.

3.2.2 Data Augmentation with BC Learning

BC learning is a novel approach that improves the performance of CNNs for image classification [6]. In our semi-supervised learning method, we incorporate BC learning to augment the training data. The BC learning method can be summarized as follows.

$$X = r\,X_1 + (1 - r)\,X_2 \tag{3.1}$$

$$Y = r\,Y_1 + (1 - r)\,Y_2 \tag{3.2}$$

Here, X_1 and X_2 are original image samples, Y_1 and Y_2 are one-hot label vectors of the samples, and r is always a random ratio sampled from a uniform distribution of [0, 1] in the whole training. The pair of X and Y is a new augmented sample for training.

Unlike other data augmentation methods that focus on the vicinity of individual samples, BC learning generates samples between different samples, enabling modeling of feature distributions across classes. The constraints in Eq. (3.1) and Eq. (3.2) simultaneously decrease the distance within the same class and increase the distance between different classes in the feature distributions. Additionally, the positional relationship among feature distributions is regulated, ensuring that the between-class samples do not cluster around decision boundaries of other classes. By modeling these feature distributions, models trained with BC learning demonstrate improved generalization and achieve better performance.

In our method, we generate samples between different classes and within the same class. This is a departure from the original setting of BC learning, which achieved the best performance with samples strictly between different classes. However, in our semi-supervised learning method, we believe that this setting better accommodates training sets that may contain some noise. To prevent further contamination of the augmented training sets, we employ a sampling strategy where either X_1 or X_2 is sampled exclusively from the labeled data. It is important to note that BC learning is only utilized for training CNNs since SVMs cannot be trained with between-class labels.

3.2.3 Experiments

3.2.3.1 Implementation Details

Figure 3.2 illustrates the architecture of the three CNNs. All networks were trained using the Stochastic Gradient Descent (SGD) optimizer. The initial learning rate was set to 0.001. During training, if the average loss of the last ten epochs decreased by less than 0.01 compared to the previous ten epochs, the learning rate was reduced to 0.0001. If the loss decline became less than 0.001, training was stopped. For CNNs without BC learning, cross-entropy loss was used, while for CNNs with BC learning, Kullback-Leibler (KL) divergence loss was employed. All CNNs were trained with a mini-batch size of 64. In the tri-training process, two iterations were performed.

3.2.3.2 Dataset and Pre-processing

We conducted experiments using 815 gastric X-ray images obtained from The University of Tokyo Hospital. The images had a resolution of 2048×2048 pixels and were from different patients. For training, we used 200 images, consisting of 100 gastritis images and 100 non-gastritis images. The remaining 615 images, comprising 140 gastritis images and 475 non-gastritis images, were used for testing. The X-ray images are divided into a number of 35×35 patches with a resolution of 299×299 pixels with a stride of 50 pixels on both width and height. The patches from annotated images are categorized as gastritis patches, non-gastritis patches and irrelevant patches. The irrelevant patches are outside the stomach and do not affect on diagnosis of gastritis/non-gastritis. Specifically, the patches from annotated gastritis images are categorized as either gastritis patches or irrelevant patches, and the patches from annotated non-gastritis images are categorized as either non-gastritis patches or irrelevant patches. As a result, the patches cropped from the 200 images of training data include 45,127 gastritis patches, 42,785 non-gastritis patches, and 48,385 irrelevant patches. Figure 3.3 shows some examples of the gastric X-ray images. In Fig. 3.3, (a) and (b) are gastritis images and (c) and (d) are non-gastritis images.

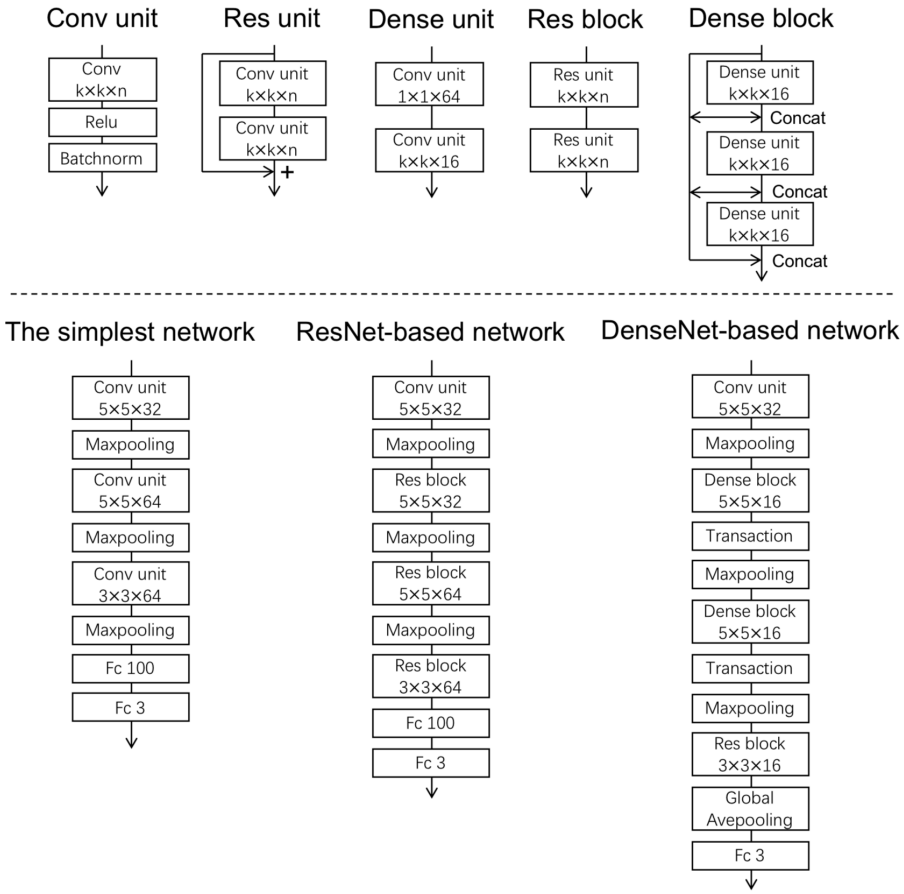
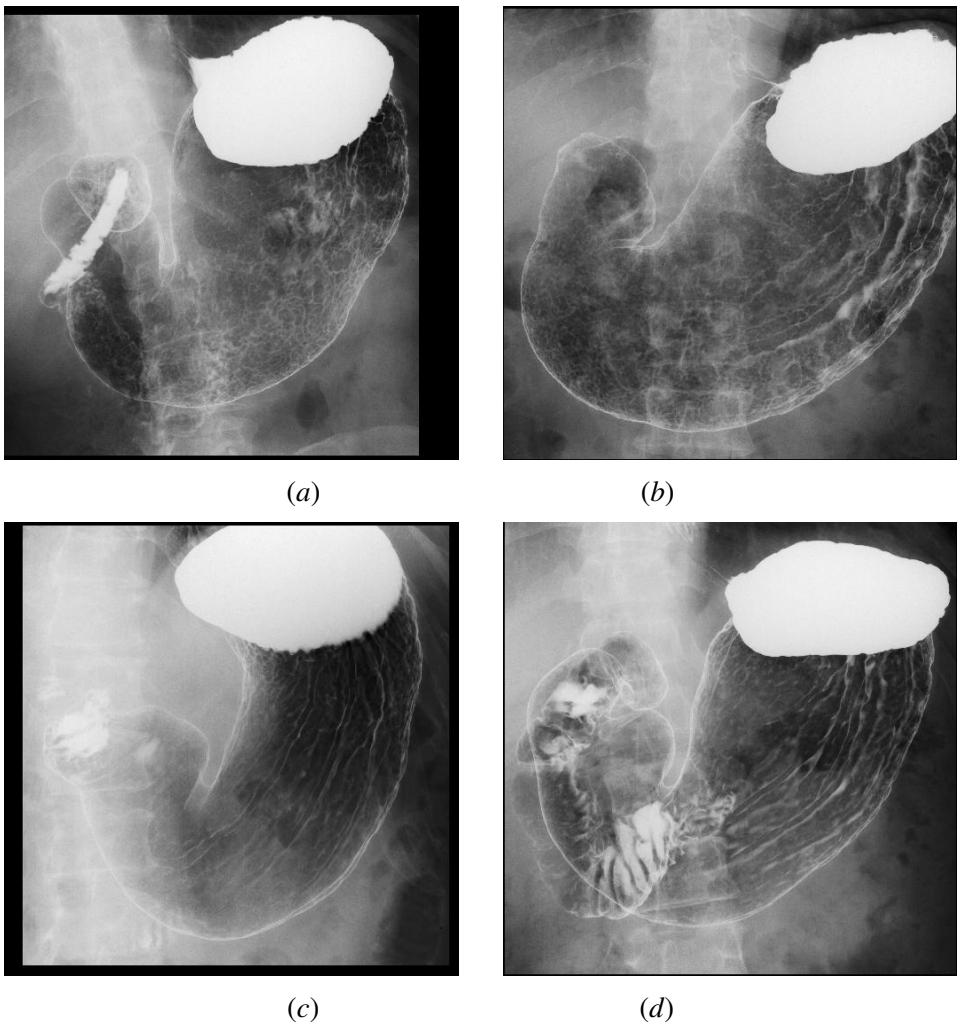


Figure 3.2: Architectures of three CNNs used in the proposed method.



(*a*) (*b*)

(*c*) (*d*)

Figure 3.3: Examples of gastric X-ray images: (a) and (b) are gastritis images and (c) and (d) are non-gastritis images.

3.2.3.3 Evaluation Method and Metrics

We evaluate gastric X-ray images by using all three models to classify patches within the image. The predictions of the models are averaged to determine the final prediction for each patch. Image categorization is then done through majority voting among patches predicted as gastritis or non-gastritis, considering only highly confident patches to improve accuracy. Confidence is measured by predicted probabilities, with a threshold of 0.7 used for stability. We directly use the original patches for evaluation, without employing BC learning. Evaluation metrics include sensitivity, specificity, and their harmonic mean. The metrics are defined as follows:

$$\begin{aligned} &\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \\ &\text{Harmonic mean} = \frac{2 \times \text{Sensitivity} \times \text{Specificity}}{\text{Sensitivity} + \text{Specificity}}, \end{aligned} \tag{3.3}$$

where TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively. In our evaluation, we consider both sensitivity and specificity as they have a trade-off relationship. To assess overall performance, we use the harmonic mean of these two metrics. In real-world clinical applications, the balance between sensitivity and specificity can be adjusted by adapting the voting system. For instance, if higher sensitivity is prioritized, the majority voting can be customized with a threshold-controlling approach, allowing gastritis patches to win with a lower count compared to non-gastritis patches.

3.2.3.4 Experimental Results

We present the results of our method using different numbers of annotated images: 10, 20, 50, 100, and 200, in Table 3.1. The convergence process of our method with 100 annotated images is shown in Table 3.2. In addition, we conducted two sets of comparison experiments. Firstly, we compared our method with a modified version that excludes BC learning, called tri-training without BC learning, to verify the effectiveness of BC learning. The results are shown in Table 3.3. Furthermore, we compared our semi-supervised learning method with supervised learning using CNNs without unlabeled data. The results of this comparison are

shown in Table 3.4. The reported results are the mean and standard deviation of multiple runs. The experiments were conducted with 3 runs for Table 3.1 and 8 runs for Table 3.2, Table 3.3, and Table 3.4. In each run, annotated images were randomly sampled from the 200 images of the training data, maintaining a 1:1 ratio between gastritis and non-gastritis images. In the case of 200 annotated images, the models were trained using supervised learning with BC learning, rather than tri-training.

Our method demonstrated high diagnostic performance for chronic gastritis, even with a small number of annotated images, as shown in Table 3.1. Remarkably, our semi-supervised learning method achieved comparable performance to supervised learning with only half of the annotated images. Moreover, increasing the number of annotated images further improved the performance. The convergence process with 100 annotated images is presented in Table 3.2, showing no significant performance improvement after the second iteration.

Table 3.3 and Table 3.4 present results using 100 annotated images, and we believe the conclusions drawn from different numbers of annotated images will remain consistent. As indicated in Table 3.3, our method, tri-training with BC learning, outperformed its ablated version without BC learning when using 100 annotated images. This confirms that data augmentation with BC learning significantly enhances the performance of our semi-supervised learning method. Furthermore, as seen in Table 4.4, semi-supervised learning surpassed supervised learning when using 100 annotated images, indicating the benefits of incorporating unlabeled images. Additionally, BC learning also improved the performance of supervised learning.

Table 3.1: Performances of our method using different numbers of annotated images.

Number of annotated images	Sensitivity	Specificity	Harmonic mean
10	0.857	0.864	0.860 ± 0.025
20	0.855	0.889	0.870 ± 0.016
50	0.871	0.945	0.906 ± 0.016
100	0.922	0.907	0.914 ± 0.001
200	0.893	0.953	0.922 ± 0.001

Table 3.2: Convergence process of our method when using 100 annotated images.

Model	Sensitivity	Specificity	Harmonic mean
SVM	0.763	0.967	0.852 ± 0.022
CNN (first iteration)	0.892	0.910	0.901 ± 0.013
CNN (second iteration)	0.915	0.914	0.914 ± 0.009

Table 3.3: Comparison of tri-training with and without BC learning using 100 annotated images.

Method	Sensitivity	Specificity	Harmonic mean
Tri-training with BC learning (our method)	0.915	0.914	**0.914 ± 0.009**
Tri-training without BC learning	0.812	0.963	0.880 ± 0.028

Table 3.4: Comparison of our method and supervised learning using 100 annotated images.

Method	Sensitivity	Specificity	Harmonic mean
Our method	0.915	0.914	**0.914 ± 0.009**
Supervised learning with BC learning	0.891	0.919	0.903 ± 0.013
Supervised learning without BC learning	0.798	0.951	0.873 ± 0.018

3.3 Conclusion

In this chapter, we have proposed a semi-supervised learning method for chronic gastritis classification using gastric X-ray images. Our method incorporates a tri-training architecture and utilizes BC learning for data augmentation. The problem of the general pseudo-label learning is solved with the tri-training architecture in our method. The outstanding performance achieved by our method demonstrates its effectiveness and potential for practical applications.

Chapter 4

Mitigation of Label Dependence with Unsupervised Domain Adaptation

4.1 Introduction

This chapter proposes several unsupervised domain adaptation (UDA) methods for mitigating the label dependence in two computer vision tasks, object detection and semantic segmentation. Specifically, in Section 4.2, we propose an adversarial learning-based method for UDA of object detection. As to UDA of semantic segmentation, we propose three methods, including a method that performs symmetric adaptation, a method based on the variational autoencoder [8], and a method that learns intra-domain style-invariant representation, in Section 4.3, Section 4.4, and Section 4.5, respectively. All the methods proposed in this chapter solve one of the problems clarified in Section 2.5 respectively.

4.2 Unsupervised Domain Adaptation of Object Detection Based on Divergence-guided Feature Alignment

4.2.1 Introduction

A prevalent approach for UDA of object detection is to perform global feature alignment with adversarial learning [7]. In the previous method of global feature alignment, domain adversarial learning is performed between an object detector and a domain discriminator. The

domain discriminator predicts domain labels for the feature maps extracted from the detector backbone, while the backbone is trained to fool the discriminator. However, this simple feature alignment approach overlooks important feature information that significantly impacts the alignment process: foreground/background and well-aligned/poorly-aligned regions. Firstly, aligning background features, which do not contain categorical information, is much less meaningful for knowledge transfer compared to aligning foreground features. Thus, the awareness of foreground/background regions becomes intuitively valuable for effective feature alignment. Additionally, being aware of well-aligned/poorly-aligned regions allows for adaptive adjustment of the alignment process.

In this section, we present a novel approach for UDA of object detection, aiming to address the aforementioned challenge. Our method focuses on adapting one-stage object detectors, an area that has received limited attention so far. We introduce a guidance mechanism based on divergence maps to facilitate the feature alignment with adversarial learning. To identify cues related to foreground regions and poorly-aligned regions in the target domain, we employ pixel-level adaptation to translate target-domain images into the source domain. By comparing the classification results between source-like images and the original target-domain images, we calculate divergence maps. The divergence maps serve as attention maps to guide the feature alignment process by spatially weighting the losses of the discriminator. We assume that foreground regions are emphasized in the divergence maps, and poorly-aligned features exhibit larger prediction divergence compared to well-aligned features. By incorporating this information, our method enables the feature alignment to effectively perceive the foreground regions and poorly-aligned regions, leading to improved adaptation performance.

4.2.2 Preliminaries

4.2.2.1 Problem Definition

Let $\mathcal{D}s = (x_i^s, y_i^s)i = 1^{n_s}$ denote the source domain, comprising n_s images x_i^s and their corresponding object annotations y_i^s. Similarly, let $\mathcal{D}t = x_i^t i = 1^{n_t}$ denote the target domain, which contains n_t unlabeled images x_i^t and shares the same object categories as $\mathcal{D}_s$. Our objective is

to train an object detector that exhibits strong generalization capabilities to the target domain by leveraging the label knowledge obtained from the source domain.

4.2.2.2 Base Object Detector

In this section, we develop our approach using the fully convolutional one-stage object detector (FCOS) [69], which has demonstrated excellent performance in one-stage object detection tasks. It's worth noting that our method is not limited to FCOS and can be applied to other popular one-stage detectors, such as SSD [70] and RetinaNet [71], as our approach does not rely on any specific characteristics unique to FCOS. Here, we provide a brief overview of the FCOS architecture.

The FCOS architecture comprises a backbone network denoted as B, which utilizes the feature pyramid network (FPN) [72] to generate feature maps at different levels. Additionally, FCOS includes a detection head denoted as H, responsible for object prediction on these feature maps. Unlike anchor-based detectors such as Faster R-CNN [2], SSD [70], and RetinaNet [71], FCOS eliminates the use of predefined anchor boxes and directly predicts the bounding boxes for objects at each location. The detection head is composed of three branches: a classification branch for object category prediction, a regression branch for bounding box regression, and a center-ness branch for filtering out low-quality bounding boxes. The center-ness branch is trained to estimate the distance between a location and the center of the corresponding object. Bounding boxes predicted at locations far from the object center are considered low-quality and are down-weighted during the non-maximum suppression (NMS) process. In the training of FCOS, three losses are utilized: a classification loss, a bounding box regression loss, and a center-ness loss. More detailed information regarding these losses can be found in the reference [69].

4.2.2.3 Global Feature Alignment

The previous approach for global feature alignment performs domain-adversarial learning between the backbone network and a discriminator, aiming to minimize the distribution gap

between the feature representations of the two domains. In the case of the FCOS object detector, global feature alignment can be achieved using the following procedure.

Given the multi-level feature maps $\{B_k(x)\}_{k=1}^{N_l}$ extracted from the FCOS backbone network B for an image x, where N_l represents the number of feature levels, a discriminator D_k ($k = 1, 2, \ldots, N_l$) is trained to predict domain labels for each feature level. The backbone network B and discriminator D_k are connected through a Gradient Reversal Layer (GRL) [73]. The GRL reverses the gradients derived from the domain classification loss and back-propagates these reversed gradients towards the backbone network B. Consequently, the backbone features $B_k(x)$ become progressively domain-invariant through the adversarial training between B and D_k. The domain classification loss for global feature alignment is defined as follows:

$$\mathcal{L}_{\text{glo}} = \sum_{k=1}^{N_l} [\mathbb{E}_{x^s \sim \mathcal{D}_s} \log D_k(B_k(x^s)) + \mathbb{E}_{x^t \sim \mathcal{D}_t} \log(1 - D_k(B_k(x^t)))]. \tag{4.1}$$

Note that D_k generates spatial results for each location on the feature maps. However, for simplicity, the spatial dimensions are omitted here. The training process is to optimize the following objective function:

$$\min_{B,H} \max_{D} \mathcal{L}(B, H, D) = \mathcal{L}_{\text{fcos}}(B, H) + \lambda \mathcal{L}_{\text{glo}}(B, D), \tag{4.2}$$

where λ is a trade-off parameter.

4.2.3 Divergence-guided Feature Alignment

4.2.3.1 Motivation

The existing global feature alignment method discussed in Section 4.2.2.3 lacks attention differentiation among different regions in an image, which can result in suboptimal alignment. This is attributed to two key factors: (1) foreground features carry more crucial information for domain adaptation than background features, and (2) within an image, there are both well-aligned and poorly-aligned regions. The first factor is evident as the objective is to transfer label knowledge for foreground objects. The second factor leads to inadequate alignment for poorly-aligned features, as the training process tends to be dominated by well-aligned objects,

which are usually more abundant among the target objects. To address these challenges, we propose a divergence-based guidance mechanism that prioritizes alignment for foreground regions and poorly-aligned regions on the feature maps. Figure 4.1 shows an overview of the proposed method.

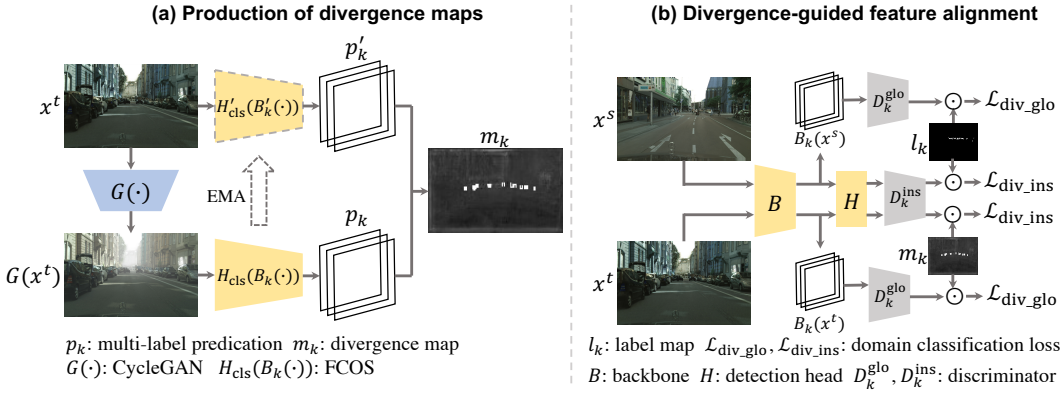


Figure 4.1: An overview of the proposed method. Note that the feature alignment is performed individually for feature maps at multiple levels ($k = 1, 2, \ldots, N_l$). However, the figure only displays one level for simplicity and clarity.

4.2.3.2 Production of Divergence Maps with Pixel-level Adaptation

Pixel-level adaptation techniques, such as image-to-image translation models like CycleGAN [29], have been effective in reducing visual differences between domains for domain adaptation. However, domain translation can be challenging in real-world scenarios and may only lead to limited improvements in detection performance. Instead of relying on pixel-level adaptation to mitigate the domain shift, we propose a novel approach that leverages divergence maps to identify cues for foreground regions and poorly-aligned regions on the feature maps. The divergence maps measure the prediction divergence between images before and after translation. Our conception is based on the observation that predictions for foreground objects are more likely to be influenced by the translation process compared to background contents, as foreground features contain richer categorical information. Additionally, features of poorly-aligned regions are domain-specific and therefore more susceptible to the effects of

translation compared to the domain-invariant features of well-aligned regions. Consequently, foreground regions and poorly-aligned regions exhibit higher prediction divergence and are highlighted in the divergence maps.

In Figure 4.1 (a), we generate divergence maps for only target-domain images. For source-domain images, we directly employ the ground-truth label maps as indicators of foreground regions. To generate the divergence maps, we begin by translating the target-domain image x^t to the source domain using a pre-trained CycleGAN model G. Subsequently, we obtain the classification outputs for x^t and $G(x^t)$ as follows:

$$p'_k = H'_{\text{cls}}(B'_k(x^t)), p_k = H_{\text{cls}}(B_k(G(x^t))), \tag{4.3}$$

where $H_{\text{cls}}(\cdot)$ is the multi-label prediction generated by the classification branch of H. Additionally, $H'_{\text{cls}}(B'k(\cdot))$ denotes a self-ensembling model of $H\text{cls}(B_k(\cdot))$. The self-ensembling model, updated after each iteration as an exponential moving average (EMA) of the detector, facilitates more stable and reliable predictions. Using the classification outputs p_k and p'_k, we calculate the divergence map using the following equation:

$$m_k^{(h,w)} = \sum_{i=1}^{N_c} ||p_k^{(h,w,i)} - p_k'^{(h,w,i)}||_2^2, \tag{4.4}$$

where h, w, i are indices for the spatial dimensions and category dimension, respectively, and N_c is the number of foreground object categories. The resulting m_k is then normalized to the range (0, 1). For any source-domain image x_s, we generate a label map l_k ($k = 1, 2, \ldots, N_l$), wherein locations labeled as positive for any object category are assigned a value of 1, and 0 otherwise. Both m_k and l_k are further clipped with a minimum value of 0.05 to prevent complete elimination of alignment for background regions.

Figure 4.2 illustrates an example of the resulting divergence maps generated by our method. The divergence maps effectively highlight the objects present in the target image across different feature levels. The left divergence map specifically emphasizes small-sized objects, as it is computed based on predictions from low-level feature maps. Similarly, the middle and right divergence maps highlight medium-sized and large-sized objects, respectively. The example shown in Fig. 4.2 demonstrates the ability of our method to effectively emphasize foreground

regions on the divergence maps, aligning with our initial assumption. Although there are no definitive indicators for qualitative analysis of well-aligned and poorly-aligned regions, we will further validate the effectiveness of our method in highlighting poorly-aligned regions and its significance through quantitative experiments.

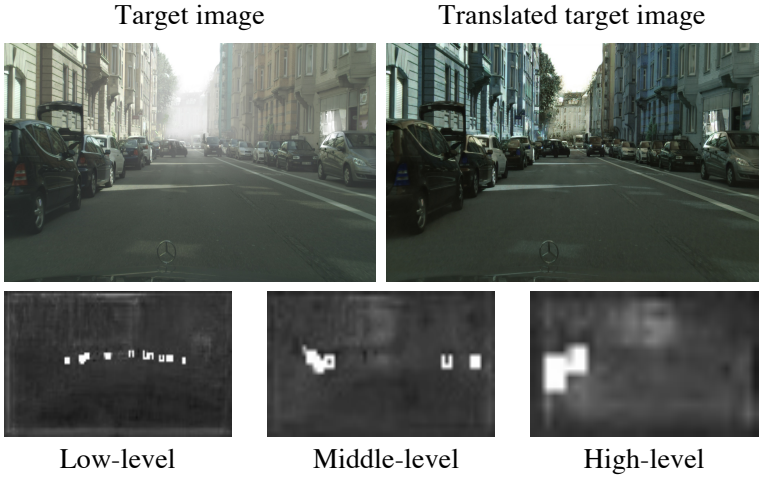


Figure 4.2: An example of the produced divergence maps. The divergence maps depicted in the figure were generated at different feature levels, specifically the first level, the third level, and the fourth level, from left to right, of the total five feature levels available.

4.2.3.3 Feature Alignment Guided with Divergence Maps at Two Levels

The feature alignment process can be guided by the divergence maps m_k and label maps l_k to highlight the alignment for foreground regions and regions with poor alignment. This is illustrated in Fig. 4.1 (b). To be more specific, we incorporate m_k and l_k as weights for the domain classification loss in Eq. (4.1), based on the feature alignment method described in Section 4.2.2.3. The loss for our feature alignment guided by divergence is defined as follows:

$$\mathcal{L}_{\text{div\_glo}} = \sum_{k=1}^{N_l} [\mathbb{E}_{x^s \sim \mathcal{D}_s} l_k \odot \log D_k(B_k(x^s)) + \mathbb{E}_{x^t \sim \mathcal{D}_t} m_k \odot \log(1 - D_k(B_k(x^t)))], \tag{4.5}$$

where "$\odot$" denotes the element-wise product. $\mathcal{L}_{\text{glo}}$ in Eq. (4.2) is replaced by $\mathcal{L}_{\text{div\_glo}}$ in the objective function of our method.

In contrast to two-stage detectors that rely on RoIPool [74] for extracting instance-level features, FCOS takes a different approach by using a detection head comprising only convolutional layers to detect objects on the backbone feature maps. This allows us to obtain instance-level feature maps that have the same spatial size as the backbone feature maps. These instance-level feature maps are generated by concatenating the outputs of the second last convolutional layers from both the classification branch and the regression branch. Additionally, the divergence-guided alignment that is applied to the backbone features can also be performed on the instance-level features in a similar manner. Since the feature alignment at the instance level complements the alignment at the backbone level, we combine them by jointly optimizing the final objective function, which is defined as follows:

$$\min_{B,H} \max_{D^{\text{glo}},D^{\text{ins}}} \mathcal{L}(B,H,D^{\text{glo}},D^{\text{ins}}) = \mathcal{L}_{\text{fcos}}(B,H)+\lambda(\mathcal{L}_{\text{div\_glo}}(B,D^{\text{glo}})+\mathcal{L}_{\text{div\_ins}}(B,D^{\text{ins}})), \quad (4.6)$$

where D^{glo} and D^{ins} denote the discriminators for backbone features and instance-level features respectively, and $\mathcal{L}_{\text{div\_ins}}$ is defined in a similar manner to $\mathcal{L}_{\text{div\_glo}}$, replacing $B_k(\cdot)$ with the instance-level features.

4.2.4 Experiments

4.2.4.1 Implementation Details

ResNet-50 [1] was employed as the backbone network for FCOS, and the discriminators comprise four 3×3 convolutional layers. For the discriminators associated with the largest and second largest feature maps, the outputs underwent downsampling by one and two convolutional layers respectively, with a stride of 2. The weight maps were downsampled to match the output size using max-pooling. During training, a stochastic gradient descent (SGD) optimizer was used for 24,000 iterations, employing an initial learning rate of 0.01 and a mini-batch size of 8 images. At iteration 18,000, the learning rate was reduced to 0.001. In the normal-to-foggy scenario, the scale parameter of the GRL was set to 0.1, and the loss weight λ was set to 1.0. In the other scenarios, the scale parameter of GRL was 0.01, and the loss weight λ was 0.1.

4.2.4.2 Datasets and Adaptation Scenarios

- *Normal-to-Foggy.* In this scenario, we used the datasets Cityscapes [4] and Foggy Cityscapes [75], where Cityscapes serves as the source domain and Foggy Cityscapes as the target domain. Cityscapes is a dataset that focuses on street scenes, comprising 2,975 images for training and 500 images for validation. All the images in Cityscapes were captured under clear weather conditions. On the other hand, Foggy Cityscapes is a derivative of Cityscapes, consisting of synthetic images depicting foggy conditions. The evaluation of the model was performed using the validation set of Foggy Cityscapes. The object categories within both datasets include “person”, “rider”, “car”, “truck”, “bus”, “train”, “motorcycle”, and “bicycle”.

- *Cross-Camera.* In this scenario, we used the datasets KITTI [76] and Cityscapes, where KITTI serves as the source domain and Cityscapes as the target domain. Although both datasets focus on street scenes, KITTI was captured using a distinct camera setup compared to Cityscapes. The KITTI dataset consists of 7,481 images for training. Evaluation was conducted on the validation set of Cityscapes for five commonly encountered categories, namely “person”, “rider”, “car”, “truck”, and “train”.

- *Synthetic-to-Real.* In this scenario, we used the datasets Synscapes [77] and Cityscapes, where Synscapes serves as the source domain and Cityscapes as the target domain. Synscapes is a synthetic dataset comprising 25,000 photo-realistic street scene images. For evaluation, the validation set of Cityscapes was used, focusing on five common categories: “person”, “car”, “truck”, “bus”, and “train”.

4.2.4.3 Methods for Comparison

For comparisons, we conducted experiments involving three existing methods that are most closely related to our method. These methods also employ adversarial learning to align the features extracted from the backbone network. The first method, referred to as Global feature alignment [7], as described in Section 4.2.2.3, aligns the backbone features without any addi-

tional information. The second method, Center-aware alignment [78], incorporates the output of the center-ness branch of FCOS into the feature alignment process to emphasize foreground features. The third method, Uncertainty-aware alignment [79], employs prediction uncertainty to assess the alignment quality and prioritizes the alignment of poorly-aligned features. In contrast, our proposed method takes into account both foreground features and poorly-aligned features through the utilization of a divergence-based guidance mechanism. It is worth mentioning that since the previous methods [7, 79] were originally designed for Faster R-CNN, we selectively retained only the alignment process for backbone features to adapt these methods for FCOS.

Table 4.1: Adaptation performance of the proposed method and previous methods in three scenarios. $C \rightarrow F$: Cityscapes to Foggy Cityscapes. $K \rightarrow C$: KITTI to Cityscapes. $S \rightarrow C$: Synscapes to Cityscapes.

Method	mAP		
	$C \rightarrow F$	$K \rightarrow C$	$S \rightarrow C$
Source only	41.1	18.2	33.3
Global feature alignment [7]	49.9	19.0	36.7
Center-aware alignment [78]	50.6	19.8	38.3
Uncertainty-aware alignment [79]	50.8	20.6	37.6
Ours ($\mathcal{L}_{\text{div\_glo}}$ only)	51.9	21.2	38.7
Ours ($\mathcal{L}_{\text{div\_glo}}$+$\mathcal{L}_{\text{div\_ins}}$)	**52.2**	**21.4**	**38.8**

4.2.4.4 Experimental Results

The adaptation performance of the proposed method and three previous methods is presented in Table 4.1. Baseline results, obtained by training solely with source-domain data, are also reported for comparison. Mean average precision (mAP) is used as the evaluation metric for the common categories, with an intersection over union (IoU) threshold of 0.5.

As shown in Table 4.1, all the domain adaptation methods showed an improvement in detection performance compared to the baseline of training solely with source data across the three adaptation scenarios. Notably, the methods [78, 79] that incorporate additional guid-

ance information for adaptation outperformed the general alignment method [7]. Remarkably, our method, which solely conducts feature alignment at the backbone level ("$\mathcal{L}_{\text{div\_glo}}$ only" in Table 4.1), achieved superior performance compared to all the previous methods that also conduct feature alignment solely at the backbone level. This finding serves as evidence for the effectiveness of guiding feature alignment with the divergence maps. Furthermore, by jointly aligning features at both the backbone level and the instance level ("$\mathcal{L}_{\text{div\_glo}}$+$\mathcal{L}_{\text{div\_ins}}$" in Table 4.1), a slight improvement in performance was observed. This suggests the complementary nature of feature alignment at the two levels.

4.2.5 Conclusion

In this section, we have presented an unsupervised domain adaptation approach specifically designed for one-stage cross-domain object detection. Our method incorporates divergence maps derived from target image predictions to guide the process of feature alignment. This unique guidance mechanism enables our method to be attentive to both foreground regions and poorly-aligned regions, leveraging this information to prioritize alignment for these specific regions. Prior works have not explored this aspect, making our method a novel contribution in the field. Experimental results in three representative adaptation scenarios validate the superiority of our method compared to the previous methods.

4.3 Unsupervised Domain Adaptation of Semantic Segmentation Based on Symmetric Adaptation Consistency

4.3.1 Introduction

For UDA of semantic segmentation, the technologies of image-to-image translation, adversarial learning, and pseudo-label learning have been proved to be effective in the previous works, and a combination of the three technologies can achieve the state-of-the-art performance. Specifically, the previous method [49] first trains an image-to-image translation model to reduce the domain shift at the image level, then trains the segmentation model with the

adversarial learning which performs the domain adaptation at the feature level, and finally re-trains the segmentation model with the pseudo labels produced with the adapted model. The key to promoting the UDA performance is the quality of the pseudo labels which is improved by the image-to-image translation and the adversarial learning in the previous method.

To further improve the pseudo-label learning, in this section, we propose a novel method based on symmetric adaptation consistency. Specifically, we adopt a symmetrical training scheme where two segmentation models are trained using adversarial learning. The key aspect of this training scheme is calculating the pixel-wise cosine similarity between the predictions of these two models in the target domain. The term "symmetrically" indicates that the adversarial losses for both models are computed separately using the source domain and the target domain. This similarity, referred to as symmetric adaptation consistency, is a crucial component. We argue that predictions exhibiting higher consistency are more reliable because the ensemble of symmetrically trained models is inherently more robust than a single model. Therefore, by leveraging symmetric adaptation consistency, we can effectively filter out noise present in the pseudo labels. By applying a threshold based on the average of symmetric adaptation consistency and the probability values, we can obtain more accurate pseudo labels compared to using only the probability values.

4.3.2 Overall Architecture

In the UDA setting, we are presented with a dataset $\mathcal{S}$ from the source domain, containing segmentation labels $\mathcal{Y}_{\mathcal{S}}$, and a dataset $\mathcal{T}$ from the target domain, which lacks any labels. The objective is to train a semantic segmentation model utilizing both $\mathcal{S}$ and $\mathcal{T}$ in order to achieve comparable performance in the target domain as in the source domain.

Our method consists of two stages: image-to-image translation and feature-level domain adaptation. Initially, images from the source domain undergo translation to resemble the target domain using an image-to-image translation model. Subsequently, the semantic segmentation model is trained using a combination of adversarial learning and pseudo-label learning. An overview of our method is illustrated in Fig. 4.3.

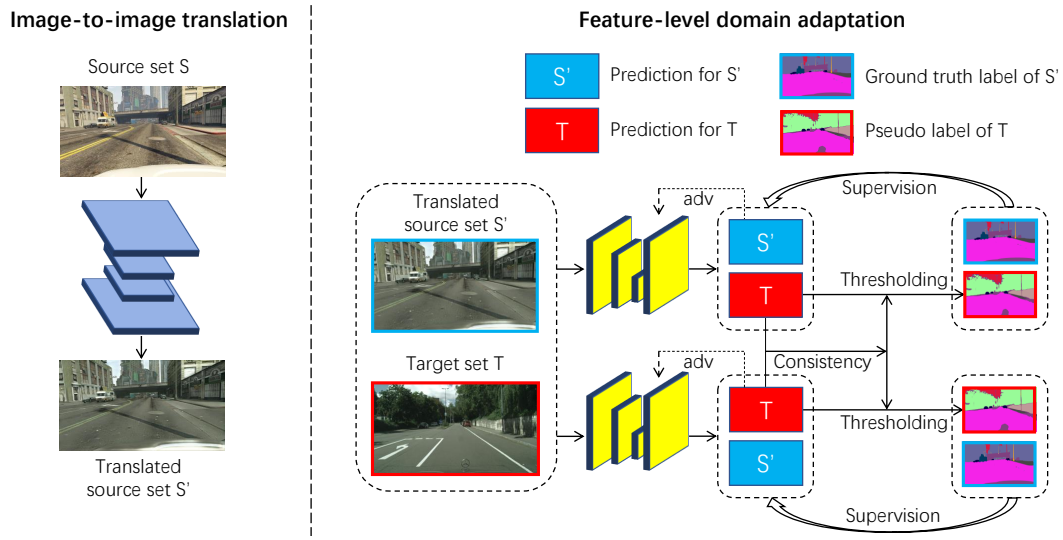


Figure 4.3: Overview of the proposed method. The "adv" in the right of the figure denotes adversarial learning.

4.3.3 Image-to-image Translation with StarGAN

The purpose of the image-to-image translation is to minimize the visual differences between the two domains, effectively performing domain adaptation at the pixel level. This process helps alleviate the challenges faced during feature-level domain adaptation. Our image-to-image translation model is constructed based on the StarGAN framework [80]. StarGAN is a generative adversarial network consisting of a generator and a discriminator. In our method, we train the translation model to establish a bidirectional mapping between the source and target domains. By mapping between the domains, mismatches in characteristics such as saturation and texture can be mitigated. The generator, denoted as G, takes an image x from either domain and a conditional label $c \in$ {Source, Target}, producing the translated image $G(x,c)$ in the domain c. On the other hand, the discriminator, denoted as D, receives the original image x and the generated image $G(x,c)$ as inputs. It outputs two terms: an adversarial term, denoted as D_{adv}, representing the probability distribution of whether the input is a real image or a fake image generated by G, and a classification term, denoted as D_{cls}, representing the probability distribution of domain classification. In the following sections, we will introduce

the respective loss functions associated with these components.

Adversarial loss To train the generator G and the discriminator D in an adversarial manner, we employ the adversarial loss. The objective is for the discriminator to distinguish between the images generated by G and real images, while G aims to generate images that are indistinguishable from real ones. The adversarial loss is defined as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_x \left[-\log D_{adv}(x)\right] + \mathbb{E}_{(x,c)} \left[-\log \left(1 - D_{adv}(G(x,c))\right)\right], \tag{4.7}$$

where the generator maximizes the latter term while the discriminator minimizes both.

Domain classification loss To train the generator G to generate images in a specific domain c, we utilize the domain classification loss. This loss is used to train G using the generated images and D using the real images. The objective of the discriminator D is to correctly classify the domain label $\hat{c}$ of the real image x. This is achieved by minimizing the following classification loss:

$$\mathcal{L}_{cls}^{D} = \mathbb{E}_{(x,\hat{c})} \left[-\log D_{cls}(\hat{c}|x)\right], \tag{4.8}$$

where $D_{cls}(\hat{c}|x)$ is the probability distribution over domain labels predicted by D. G is trained using the generated images by minimizing the following classification loss for G:

$$\mathcal{L}_{cls}^{G} = \mathbb{E}_{(x,c)} \left[-\log D_{cls}(c|G(x,c))\right], \tag{4.9}$$

and G can consequently generate images that are classified as the specified domain c by D.

Reconstruction loss To ensure that the translated images preserve their original contents, we introduce a reconstruction constraint to the generator G. This constraint aims to reconstruct the original image x when given the translated image $G(x,c)$ and the corresponding original domain label $\hat{c}$. This is achieved by minimizing the following reconstruction loss:

$$\mathcal{L}_{rec} = \mathbb{E}_{(x,c,\hat{c})} \left[\|x - G(G(x,c),\hat{c})\|_1\right]. \tag{4.10}$$

Identical loss To enhance the preservation of original details in the translated images, we introduce an additional identical loss. The identical loss is defined as follows:

$$\mathcal{L}_{ide} = \mathbb{E}_{(x,\hat{c})} \left[\|x - G(x,\hat{c})\|_1\right], \tag{4.11}$$

which encourages G to give an output with no changes from the original image if the specified domain is the original domain.

Semantic consistency loss Given that the ultimate objective is semantic segmentation, we incorporate an extra semantic consistency loss to improve the preservation of semantic information. The semantic consistency loss is defined as follows:

$$\mathcal{L}_{sem} = \mathbb{E}_{(x,c)} \left[||M(x) - M(G(x,c))||_2 \right], \tag{4.12}$$

where M is a semantic segmentation model trained with $\mathcal{S}$.

Full objective function Combining all the above losses, we have the final loss functions for G and D as follows:

$$\mathcal{L}_D = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^D_{cls}, \tag{4.13}$$

$$\mathcal{L}_G = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^G_{cls} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{ide}\mathcal{L}_{ide} + \lambda_{sem}\mathcal{L}_{sem}, \tag{4.14}$$

where λ_{cls}, λ_{rec}, λ_{ide} and λ_{sem} are loss weights.

After training the translation model using the aforementioned loss function, we use this model to translate the source-domain dataset $\mathcal{S}$ into a new domain $\mathcal{S}'$ that is visually similar to the target domain.

4.3.4 Symmetric Feature-level Domain Adaptation

4.3.4.1 Motivation

In the feature-level domain adaptation, we align the feature distributions through adversarial learning while leveraging pseudo-label learning to enhance performance. We train a discriminator D_{da} to classify the outputs of the segmentation model M as either belonging to the source domain or the target domain. Conversely, M learns to deceive D_{da} in an adversarial manner. However, unlike traditional adversarial learning, simply opposing the adversarial loss for M to that of D_{da} is not sufficient because our objective is to align the feature distributions between the two domains. Previous studies [39,49,81] have mainly focused on adapting the feature representations of the target domain. In other words, they train M to generate outputs that are from

the target domain and are classified as the source domain by D_{da}, thereby aligning the feature distribution of the target domain with that of the source domain. However, we believe that there is another approach: adapting the features of the source domain. As shown in Fig. 4.4, both approaches involve bringing either the source or target domain closer to the other, resembling two symmetric adaptation procedures. We argue that such an architecture is more robust than unidirectional adaptation. Therefore, we propose training two models symmetrically and considering the consistency of predictions when generating the pseudo-labels.

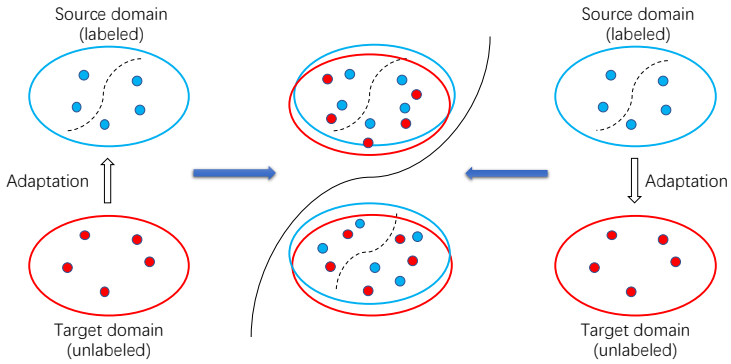


Figure 4.4: Illustration of the symmetric adaptation.

4.3.4.2 Pseudo-label Generation

We propose a symmetric training approach for two segmentation models, namely M_1 and M_2, where each model is adapted with the source domain and the target domain, respectively. These models are then used to generate pseudo labels. Averaging the predictions of both models can enhance the accuracy of the pseudo labels. However, to demonstrate that the improvement is not solely due to averaging the predictions, but rather a result of the proposed symmetric adaptation consistency, we generate two separate sets of pseudo labels using each model without performing the averaging step. To determine the confidence of the labels, we measure a confidence map $\mathcal{M}_{confi}$ by averaging the probability map $\mathcal{M}_{proba}$ and the symmetric

adaptation consistency map $\mathcal{M}_{consist}$ as follows:

$$\mathcal{M}_{confi} = \frac{\mathcal{M}_{proba} + \mathcal{M}_{consist}}{2}. \tag{4.15}$$

We exclude labels with a confidence level below 0.95 from the pseudo label sets. However, if a class has more than half of its pixels excluded, we relax the constraint and ensure that at least half of the assigned pixels for each class are included to maintain balance in the pseudo label sets. We experimentally select the threshold of 0.95, as the number of included pixels starts to decrease rapidly beyond this value. By iteratively performing the procedure of pseudo label generation and new model training twice, our method achieves the best performance.

4.3.4.3 Training Losses

Supervised learning loss in the source domain Transferring label information from the source domain to the target domain is achieved through supervised learning using the translated domain $\mathcal{S}'$. We employ the cross-entropy loss to train the segmentation model M with images $x_{\mathcal{S}'} \in \mathcal{S}'$ and one-hot labels $y_{\mathcal{S}'} \in \mathcal{Y}_{\mathcal{S}'}$. The loss can be written as follows:

$$\mathcal{L}_{seg\_s'} = \mathbb{E}_{(x_{\mathcal{S}'}, y_{\mathcal{S}'})} \left[-\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} y_{\mathcal{S}'}^{(h,w,c)} \log M(x_{\mathcal{S}'})^{(h,w,c)} \right], \tag{4.16}$$

where H, W, and C denote the height, width, and number of categories, respectively.

Pseudo-label learning loss in the target domain The subsequent pseudo-label learning loss has a similar definition to the supervised learning loss, but is applied in different domains.

$$\mathcal{L}_{seg\_t} = \mathbb{E}_{(x_{\mathcal{T}}, y_{\mathcal{T}})} \left[-\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} y_{\mathcal{T}}^{(h,w,c)} \log M(x_{\mathcal{T}})^{(h,w,c)} \right], \tag{4.17}$$

where $x_{\mathcal{T}}$ denotes the image of $\mathcal{T}$, and $y_{\mathcal{T}} \in \mathcal{Y}_{\mathcal{T}}$ denotes the pseudo label set.

Adversarial loss D_{da} and M engage in an adversarial learning process, where D_{da} aims to classify M's outputs, while M strives to generate outputs that are misclassified by D_{da}. The loss functions are formulated as follows:

$$\mathcal{L}_{adv}^{D_{da}} = \mathbb{E}_{x_{\mathcal{S}'}} \left[-\log D_{da}(M(x_{\mathcal{S}'})) \right] + \mathbb{E}_{x_{\mathcal{T}}} \left[-\log \left(1 - D_{da}(M(x_{\mathcal{T}}))\right) \right], \tag{4.18}$$

$$\mathcal{L}_{adv}^{M} = \begin{cases} \mathbb{E}_{x_{\mathcal{S}'}} \left[-\log \left(1 - D_{da}(M(x_{\mathcal{S}'}))\right) \right] & \text{if } M = M_1, \\ \mathbb{E}_{x_{\mathcal{T}}} \left[-\log D_{da}(M(x_{\mathcal{T}})) \right] & \text{if } M = M_2, \end{cases} \tag{4.19}$$

where D_{da} outputs the probability that the input belongs to the source domain. We employ two separate discriminators for M_1 and M_2, although their loss functions are identical. To simplify the presentation, we express them using a single equation.

Full objective function Equation (4.18) is the full loss function for training D_{da}, and the full loss function for M is defined as follows:

$$\mathcal{L}_M = \mathcal{L}_{seg\_s'} + \mathcal{L}_{seg\_t} + \lambda_{adv}\mathcal{L}^M_{adv}, \tag{4.20}$$

where λ_{adv} is the loss weight.

4.3.5 Experiments

4.3.5.1 Implementation Details

We adopted the network architectures and training parameters of Stargan [80] for the image-to-image translation model. As for the semantic segmentation model, we used the DeepLab V2 [3] with ResNet101 [1] architecture. The discriminator architecture and training parameters for both the segmentation model and the discriminator remained consistent with previous works [39, 49].

4.3.5.2 Datasets and Adaptation Scenarios

We conducted experiments on the widely used domain adaptation scenario, specifically the GTA5-to-Cityscapes scenario. In this setup, the GTA5 dataset [82] served as the source domain, while the Cityscapes dataset [82] served as the target domain. Here are the details of the datasets.

The Cityscapes dataset consists of real-world urban scene images with a resolution of 2,048×1,024 pixels. It comprises a training set containing 2,975 images and a validation set containing 500 images. In our experiments, we used the validation set as the test data. The images were resized to a resolution of 1,024×512 pixels.

The GTA5 dataset consists of 24,966 synthetic urban scene images collected from the GTA5 video game. The images have a resolution of 1,914×1,052 pixels. Nineteen common object

categories are shared between the GTA5 and Cityscapes datasets. The images were resized to a resolution of 1,280×720 pixels.

4.3.5.3 Experimental Results

We calculated the mean Intersection over Union (IoU) for the 19 categories and compared our results with other state-of-the-art methods, as shown in Table 4.2. To ensure a fair comparison, all the results reported in Table 4.2 were obtained using the DeepLab V2 with ResNet101 architecture. Among the methods listed in Table 4.2, BDL is the closest method to ours. They share the same components as ours, except for the symmetric adaptation and consistency. In Table 4.2, the "Single" result for our method represents the average mean IoU achieved by the two models in our method when tested individually. This result serves as an indicator of the performance of a single model in our method. The "Fusion" result is obtained by averaging the predictions of the two models. According to Table 4.2, our method achieved the highest mean IoU of 47.9 when using a single model for testing. Furthermore, by averaging the predictions of the two models, our method achieved an additional improvement of 0.6, demonstrating the robustness and effectiveness of the symmetric adaptation approach.

Table 4.2: Results of mean intersection over union (IoU) for GTA5-to-Cityscapes benchmark.

Method	mIoU	Method	mIoU
AdaptSegNet [39]	41.4	Cycada [38]	42.7
DCAN [83]	41.7	ADVENT [46]	45.5
CLAN [81]	43.2	Patch-Align [84]	46.5
DISE [85]	45.4	BDL [49]	47.2
Ours (single)	47.9	Ours (fusion)	**48.5**

To demonstrate the effectiveness of incorporating the symmetric adaptation consistency, we conducted additional ablation studies. We first compared the mean accuracy of all categories of pseudo labels in the first iteration that are obtained using and not using the consistency. When not using the consistency, the pseudo labels were filtered based solely on their probability,

while ensuring that the number of pixels for each category remained the same as when using the consistency. We also compared the final performances under these conditions. The results are presented in Table 4.3. By considering the consistency, the mean accuracy of the pseudo labels showed a notable improvement of 2.0. This improvement in the quality of the pseudo labels had a positive impact on the final performance, with the mean IoU improved by 1.3. These results provide further evidence of the effectiveness of incorporating the symmetric adaptation consistency in our method.

Table 4.3: Results of mean intersection over union (IoU) for ablation study.

Method	Accuracy of pseudo labels	mIoU
Using consistency	**72.6**	**48.5**
Not using consistency	70.6	47.2

4.3.6 Conclusion

In this section, we have presented a novel symmetric domain adaptation architecture for semantic segmentation, along with a method that incorporates the symmetric adaptation consistency to enhance the adaptation performance. Through extensive ablation studies, we have demonstrated the effectiveness of our approach. Our method outperformed previous methods in the task of domain adaptation for semantic segmentation.

4.4 Unsupervised Domain Adaptation of Semantic Segmentation Using Variational Autoencoder

4.4.1 Introduction

The adversarial learning is one of the most prevalent technologies for UDA of semantic segmentation, which is also used in the method proposed in Section 4.3 of this thesis. Despite the good performance, the optimization of the adversarial learning is always difficult and unstable

due to the minimax game between the segmentation model and the discriminator. Therefore, in this section, we propose a method in which the discriminator is replaced by a variational autoencoder (VAE) and the training is performed in a non-adversarial manner. Our method represents a pioneering attempt to leverage VAE in domain adaptation. More specifically, our method employs a VAE model to capture the output distribution of the source domain, while simultaneously updating the segmentation model to align the output distribution of the target domain with the distribution learned by the VAE model. The segmentation model and the VAE are trained simultaneously with a common loss in a non-adversarial manner, which is more stable and easier to perform than the adversarial learning. Moreover, we show that the VAE-based learning and the adversarial learning are complementary and the performance can therefore be further improved by combining the two technologies.

4.4.2 Overall Architecture

The proposed method addresses the task of learning a semantic segmentation model M that can accurately predict pixel labels for the target domain $\mathcal{T}$, despite the absence of annotations in $\mathcal{T}$, while leveraging the pixel-wise annotations available in the source domain $\mathcal{S}$. The key objective of our method is to align the feature distributions of $\mathcal{S}$ and $\mathcal{T}$. To accomplish this, we employ a variational autoencoder V, which consists of an encoder V_{enc} and a decoder V_{dec}, to learn the output distribution of the segmentation model. Unlike adversarial learning-based approaches that involve a minimax game between the segmentation model M and a discriminator, our method trains M and V with a shared objective. In addition to the feature alignment with the VAE, we introduce the adversarial learning and the pseudo-label learning to further improve the performance.

4.4.3 VAE-based Feature Alignment

Figure 4.5 illustrates our VAE-based UDA method. Our method follows a two-step training process: 1) updating V using the target domain $\mathcal{T}$, and 2) updating M using both the source and target domains while keeping V fixed. These steps are performed iteratively throughout the

training process, gradually aligning the output distributions of the two domains. We provide a detailed explanation of each step as follows.

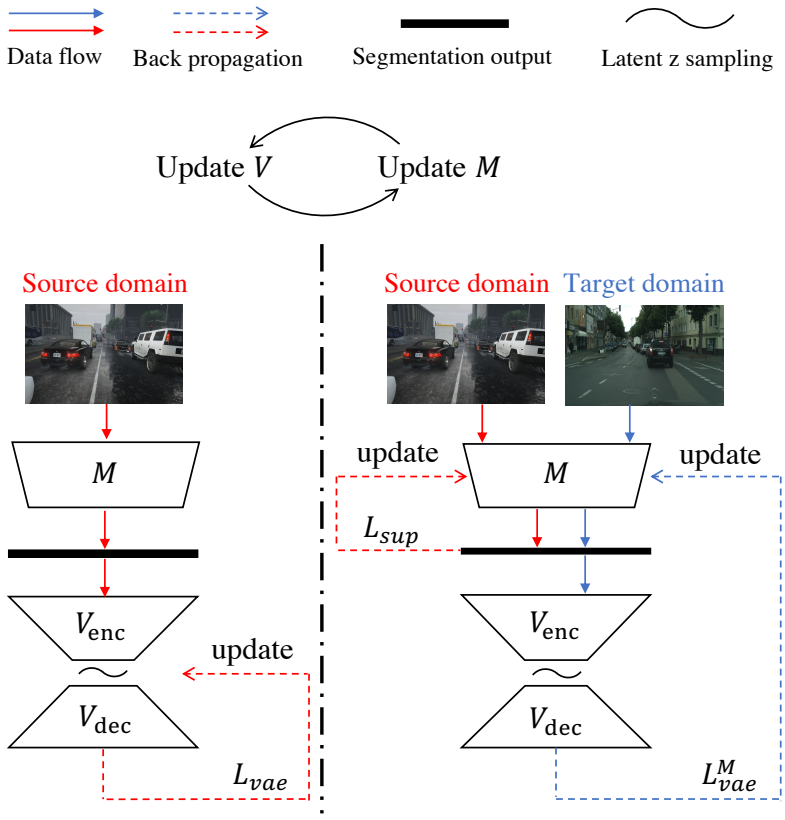


Figure 4.5: Illustration of our VAE-based UDA method. M is the segmentation model. V is the variational autoencoder composed of encoder V_{enc} and decoder V_{dec}.

4.4.3.1 Update of VAE

In our method, we employ a basic form of VAE. The VAE consists of an encoder, denoted as V_{enc}, and a decoder, denoted as V_{dec}. When given an input x from the source domain $\mathcal{S}$, the encoder V_{enc} encodes the segmentation output $y = M(x)$ into a latent distribution $V_{\text{enc}}(z|y)$, which is typically modeled as a Gaussian distribution:

$$V_{\text{enc}}(z|y) = \mathcal{N}(\mu(y), \Sigma(y)), \tag{4.21}$$

where $\mu(y)$ and $\Sigma(y)$ are the distribution parameters estimated by the encoder. To regularize the encoder, we impose a prior distribution $p(z) = \mathcal{N}(0, I)$ and minimize the Kullback-Leibler divergence $\mathrm{KL}[V_{\mathrm{enc}}(z|y)||p(z)]$. The decoder V_{dec} uses the reparameterization trick [8] to sample a latent vector z from the distribution $V_{\mathrm{enc}}(z|y)$ and reconstructs $M(x)$ as follows:

$$y' = V_{\mathrm{dec}}(z), z \sim V_{\mathrm{enc}}(z|y). \tag{4.22}$$

The reconstruction output y' is compared to the original segmentation output $M(x)$ using an L1 loss. The complete loss function for V is defined as follows:

$$\mathcal{L}_{vae} = \mathbb{E}_{x \in \mathcal{S}} \left[||y' - M(x)||_1 + \lambda_{kld} \mathrm{KL} \left[V_{\mathrm{enc}}(z|M(x))||p(z) \right] \right], \tag{4.23}$$

where λ_{kld} is the weight for the regularization term of $V_{\mathrm{enc}}(z|M(x))$. By minimizing this equation, we can effectively learn the output distribution of the source domain $\mathcal{S}$ using the VAE.

4.4.3.2 Update of Segmentation Model

The semantic features learned solely from the source domain $\mathcal{S}$ may not be as informative for the target domain $\mathcal{T}$, resulting in a significant decrease in performance when applying the learned model to $\mathcal{T}$. To address this issue, we train the segmentation model M using both domains $\mathcal{S}$ and $\mathcal{T}$. In the source domain $\mathcal{S}$, M learns semantic features using pixel-wise ground truth labels. On the other hand, in the target domain $\mathcal{T}$, M aims to minimize the VAE loss while keeping the parameters of V fixed. This objective is designed to align the output distribution of $\mathcal{T}$ with the distribution learned by V. Since V learns the output distribution of $\mathcal{S}$, this learning objective indirectly encourages consistency between the output distributions of both domains. By training with data from both domains, M acquires domain-invariant features, which helps to mitigate the performance drop when applying the model to the target domain $\mathcal{T}$.

We perform supervised learning with the source domain $\mathcal{S}$ by minimizing the cross-entropy loss as follows:

$$\mathcal{L}_{sup} = \mathbb{E}_{(x, y_{gt}) \in \mathcal{S}} \left[-\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} y_{gt}^{(h,w,c)} \log M(x)^{(h,w,c)} \right], \tag{4.24}$$

where y_{gt} denotes one-hot label, $M(x)$ denotes the predicted probability distribution, H, W and C denote the height, weight, and the number of classes, respectively.

For feature alignment, we use a loss function similar to $\mathcal{L}_{vae}$ but applied to the other domain:

$$\mathcal{L}_{vae}^{M} = \mathbb{E}_{x\in\mathcal{T}}\left[\|y' - M(x)\|_1 + \lambda_{kld}\mathrm{KL}\left[V_{\mathrm{enc}}(z|M(x))\|p(z)\right]\right]. \tag{4.25}$$

To train M, we combine these losses in a weighted sum:

$$\mathcal{L}_M = \mathcal{L}_{sup} + \lambda_{vae}\mathcal{L}_{vae}^{M}, \tag{4.26}$$

where λ_{vae} is the loss weight. By optimizing M with this loss function, we facilitate the transfer of semantic knowledge from the source domain $\mathcal{S}$ to the target domain $\mathcal{T}$.

4.4.4 Integration with Adversarial Learning and Pseudo-label Learning

As our method using VAE is compatible with the UDA method based on the adversarial learning, we incorporate an additional adversarial loss into our method. We adopt a representative adversarial learning method [39], which involves training a discriminator to distinguish the domain of the segmentation output, and simultaneously training the segmentation model to deceive the discriminator. To integrate this into our method, we introduce a discriminator D and update the segmentation model M using both $\mathcal{L}_{vae}^{M}$ and the following adversarial loss:

$$\mathcal{L}_{adv} = \mathbb{E}_{x\in\mathcal{T}}\left[-\log D(M(x))\right], \tag{4.27}$$

where $D(M(x))$ represents the probability of the segmentation output belonging to the source domain as recognized by the discriminator. The loss function for the discriminator D is defined as follows:

$$\mathcal{L}_D = \mathbb{E}_{x\in\mathcal{S}}\left[-\log D(M(x))\right] + \mathbb{E}_{x\in\mathcal{T}}\left[-\log(1 - D(M(x)))\right]. \tag{4.28}$$

Pseudo labels serve as additional supervision in the unlabeled domain $\mathcal{T}$ and greatly enhance the overall performance. To generate pseudo labels, we train the segmentation model M using both the VAE loss and the adversarial loss. Subsequently, we utilize M to predict labels for the data in $\mathcal{T}$. These predicted labels are selected based on a probability threshold, and they are utilized as pseudo labels. The training with pseudo labels is similar to supervised learning with ground truth labels. The corresponding loss function is defined as follows:

$$\mathcal{L}_{psl} = \mathbb{E}_{(x,y_{psl})\in\mathcal{T}}\left[-\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C} y_{psl}^{(h,w,c)}\log M(x)^{(h,w,c)}\right], \tag{4.29}$$

where y_{psl} is the pseudo label.

In our method, the final loss function for the segmentation model M is obtained by combining the VAE-based feature alignment, the adversarial learning, and the pseudo-label learning, as defined in the following equation:

$$\mathcal{L}_{M\_final} = \mathcal{L}_{sup} + \mathcal{L}_{psl} + \lambda_{vae}\mathcal{L}_{vae}^{M} + \lambda_{adv}\mathcal{L}_{adv}, \tag{4.30}$$

where λ_{vae} and λ_{adv} is loss weights.

4.4.5 Experiments

4.4.5.1 Implementation Details

Following our method proposed in Section 4.3, for the proposed method in this section, we use the same segmentation network, DeepLab-v2 [3] network with ResNet-101 [1]. The training parameters and the discriminator in the adversarial learning are also the same as those in Section 4.3.5.1. The VAE architecture employed in our method consists of a total of 12 convolutional layers. The encoder component is comprised of 6 convolutional layers, with the respective filter numbers being {64, 128, 256, 512, 512, 512}. Additionally, there is a linear layer that projects the encoded features into a latent space. The convolutional layers have a filter size of 4×4 and a stride of 2. On the other hand, the decoder mirrors the encoder's structure, mapping the latent vectors back to the output space of the segmentation model. The dimensionality of the latent space is set to 2,048. For the optimization of the VAE, we utilize the Adam optimizer with an initial learning rate of 10^{-3} and momentum values of 0.9 and 0.99. The learning rate follows the same scheduling policy as the segmentation model. The weight parameters λ_{vae} and λ_{kld} are set to 10^{-3} and 0.05, respectively.

4.4.5.2 Datasets and Adaptation Scenarios

We conducted experiments on two benchmark domain adaptation scenarios, GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes. GTA5 dataset [82] and SYNTHIA dataset [86] were used as the

source domain, and Cityscapes dataset [82] was used as the target domain. The datasets are described as follows.

The Cityscapes dataset is a real-world dataset comprising urban scene images with a resolution of 2,048×1,024 pixels. It consists of a training set containing 2,975 images and a validation set with 500 images. For our experiments, we used the validation set as the test data. The images were resized to a resolution of 1,024×512 pixels.

The GTA5 dataset is a collection of 24,966 synthesized urban scene images generated from the GTA5 video game. These images have a resolution of 1,914×1,052 pixels. There are nineteen common categories shared between the GTA5 dataset and the Cityscapes dataset. The images were resized to a resolution of 1,280×720 pixels.

The SYNTHIA dataset is a synthetic dataset consisting of photo-realistic images depicting various driving scenarios within a virtual city. We used the SYNTHIA-RAND-CITYSCAPES subset, which contains 9,400 images with a resolution of 1,280×760 pixels. This subset shares 16 common categories with the Cityscapes dataset.

4.4.5.3 Experimental Results

The results of various state-of-the-art methods, including our method, in the GTA5-to-Cityscapes scenario are presented in Table 4.4. The table displays the Intersection over Union (IoU) values for each class as well as the mean IoU across all classes. Note that we did not include the methods [38, 49, 87] which employ image-to-image translation models. These methods involve two stages, namely image-to-image translation and segmentation, which fall outside the scope of our comparison. Since our method incorporates both adversarial learning and pseudo-label learning, we conducted ablation studies to assess the effectiveness of each component. In Table 4.4, we have included the results for “vae-based” which solely performs VAE-based feature alignment as described in Section 4.4.3, and “adv-based” which exclusively utilizes the adversarial learning-based method [39] outlined in Section 4.4.4. As shown in Table 4.4, both “vae-based” and ‘adv-based” approaches achieved comparable performances, with our VAE-based method slightly outperforming in terms of mean IoU. By

combining the two methods (referred to as “vae+adv”), we observed an improvement in mean IoU, underscoring the complementary nature of these approaches. Furthermore, the inclusion of pseudo-label learning (indicated by “vae-based+psl” and “adv-based+psl”) resulted in enhanced mean IoUs for both the “vae-based” and “adv-based” methods. Ultimately, by integrating adversarial learning, pseudo-label learning, and VAE-based feature alignment, our method (termed “vae+adv+psl”) achieved the highest mean IoU, surpassing all other methods in the comparison.

The results in the SYNTHIA-to-Cityscapes scenario are presented in Table 4.5. Some comparative methods only reported performances for 13 classes, excluding the classes “wall”, “fence” and “pole”. Therefore, we have included the mean IoUs for both 13 classes (denoted as “mIoU*”) and 16 classes in the table. As shown in Table 4.5, our method achieved superior performance compared to most of the comparative methods in terms of both mean IoUs for 13 classes and 16 classes. However, our method did obtain a slightly lower mean IoU for 13 classes compared to SSF-DAN [88]. It is important to highlight that although SSF-DAN outperformed our method by 0.8 in terms of mean IoU for 13 classes in the SYNTHIA-to-Cityscapes scenario, our method demonstrated greater superiority over SSF-DAN in the GTA5-to-Cityscapes scenario, with a difference of 1.6 in mean IoU. The results presented in both Table 4.4 and Table 4.5 reaffirm that our method exhibits state-of-the-art performance and remarkable robustness across different adaptation scenarios.

Table 4.4: Results of intersection over union (IoU) for GTA5-to-Cityscapes benchmark.

Method	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
AdaptSegNet [39]	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
DISE [85]	91.5	47.5	82.5	**31.3**	25.6	33.0	33.7	25.8	82.7	28.8	**82.7**	62.4	**30.8**	85.2	27.7	34.5	6.4	25.2	24.4	45.4
Patch-Alignment [84]	**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
CLAN [81]	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
ADVENT [46]	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
MaxSquare [47]	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	**85.3**	39.4	78.2	**63.0**	22.9	84.6	36.4	43.0	5.5	**34.7**	33.5	46.4
SSF-DAN [88]	90.3	38.9	81.7	24.8	22.9	30.5	**37.0**	21.2	84.8	38.8	76.9	58.8	30.7	**85.7**	30.6	38.1	5.9	28.3	36.9	45.4
vae-based	83.0	26.3	79.2	21.0	21.5	28.7	28.7	20.1	82.5	38.8	76.3	53.9	21.2	75.5	32.3	34.7	11.6	28.0	31.2	41.8
adv-based	84.2	27.3	78.0	13.3	19.9	26.5	27.5	19.4	83.2	41.7	78.3	55.3	22.8	74.4	25.8	38.1	11.8	24.7	34.7	41.4
vae-based+psl	84.3	37.2	82.7	13.3	21.9	31.9	34.0	25.2	83.8	40.6	78.1	57.1	25.6	78.1	24.6	42.7	11.4	27.3	28.2	43.6
adv-based+psl	79.1	31.3	81.4	11.7	20.0	30.7	31.6	24.5	83.2	**43.2**	80.6	55.8	25.4	78.5	24.9	44.1	**17.1**	29.1	30.0	43.2
vae-based+adv-based	89.1	32.9	79.4	25.1	19.0	28.5	29.4	20.8	84.2	42.7	74.5	55.5	25.2	83.1	27.7	41.0	11.7	28.4	34.7	43.8
vae+adv+psl(our method)	91.9	44.3	**82.8**	26.8	24.6	**34.0**	34.3	26.8	84.8	42.1	80.6	57.2	27.4	81.8	22.5	43.0	14.9	31.3	**41.4**	**47.0**

Table 4.5: Results of intersection over union (IoU) for SYNTHIA-to-Cityscapes benchmark. "mIoU" is the mean IoU over all of the 16 categories, and "mIoU*" is that over 13 categories excluding 3 categories marked by "*". Results of "-" were not reported in the papers.

Method	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	sky	person	rider	car	bus	motorcycle	bicycle	mIoU	mIoU*
AdaptSegNet	79.2	37.2	78.8	-	-	-	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	-	45.9
DISE	**91.7**	**53.5**	77.1	2.5	0.2	27.1	6.2	7.6	78.4	81.2	55.8	19.2	**82.3**	30.3	17.1	34.3	41.5	48.8
Patch-Alignment [84]	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
CLAN [81]	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
ADVENT [46]	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
MaxSquare [47]	82.9	40.7	80.3	10.2	**0.8**	25.8	12.8	18.2	**82.5**	82.2	53.1	18.0	79.0	31.4	10.4	35.6	41.4	48.2
SSF-DAN [88]	84.6	41.7	**80.8**	-	-	-	11.5	14.7	80.8	**85.3**	57.5	21.6	82.0	36.0	19.3	34.5	-	**50.0**
Our method	80.6	37.2	77.3	**11.2**	0.5	**29.8**	**20.5**	**18.3**	76.0	82.7	**58.0**	23.4	67.4	29.5	**24.7**	**44.1**	**42.6**	49.2

4.4.6 Conclusion

In this section, we have proposed a VAE-based method for UDA of semantic segmentation. Our method trains a VAE to perform feature alignment in the output space of the segmentation model. Because our method performs the domain adaptation in a non-adversarial manner, it is easier to train and more stable than the adversarial learning-based methods. Moreover, our method can be combined with the adversarial learning and the pseudo-label learning for further improvement. The effectiveness of our method has been confirmed by conducting the experiments in two benchmark scenarios.

4.5 Unsupervised Domain Adaptation of Semantic Segmentation Learning Intra-domain Style-invariant Representation

4.5.1 Introduction

Previous approaches to UDA in semantic segmentation have primarily focused on reducing the differences between the source and target domains. However, an aspect that has received insufficient attention is the fact that aligning feature distributions alone does not guarantee the generalization ability of the trained model in the target domain. The distinct data distributions and nontransferable features between the two domains make complete alignment unachievable. Consequently, a model trained solely on supervision signals from the source domain may not generalize well to the target domain. In this section, we address this issue by emphasizing the importance of learning intra-domain style-invariant representations for UDA in semantic segmentation. The fundamental idea is that if the learned representation remains invariant to the diverse characteristics present in the target domain, such as brightness, saturation, and texture variations, the segmentation model will perform well on unknown samples in the target domain. Our objective is to learn this style-invariant representation by leveraging both supervised learning with labeled data from the source domain and unsupervised learning with unlabeled data from the target domain. To achieve this, we propose a self-ensembling method that integrates supervised and unsupervised learning to obtain the intra-domain style-invariant

representation.

As mentioned earlier, the presence of diverse intra-domain styles in images is crucial for realizing our method. In our method, we achieve this by performing style translation on the source-domain images to different target domain styles, while also diversifying the styles of the target-domain images. To accomplish this task, we employ an existing multimodal unpaired image-to-image (I2I) translation technique called multimodal unsupervised image-to-image translation (MUNIT) [89]. However, during our experimentation, we encountered a crucial requirement that the existing method failed to meet. Specifically, we needed to ensure the consistency of semantic content in the translated results. To address this challenge, we propose a semantic-aware MUNIT, which modifies the MUNIT architecture to enable content-consistent translation. This is achieved by incorporating pixel-level semantic information as additional guidance for the translation process.

4.5.2 Learning Intra-domain Style-invariant Representation with Self-ensembling

4.5.2.1 Overall Architecture

To transfer semantic knowledge at the pixel level from a labeled source-domain dataset $\mathcal{S}$ (consisting of image-label pairs $\{x^s, y^s\} \in \mathcal{S}$) to an unlabeled target-domain dataset $\mathcal{T}$ (containing images $\{x^t\} \in \mathcal{T}$), we propose a self-ensembling method that aims to learn intra-domain style-invariant representations. The self-ensembling architecture comprises two models: a student model M and a teacher model M' that have identical structures. The student model is trained simultaneously using both labeled data from $\mathcal{S}$ and unlabeled data from $\mathcal{T}$. Meanwhile, the teacher model is updated as an exponential moving average (EMA) of the student model. This update is governed by the equation:

$$\theta'_k = \alpha\theta'_{k-1} + (1-\alpha)\theta_k, \tag{4.31}$$

where α denotes the EMA weight parameter, θ'_k denotes the weights of M' at training step k, θ'_{k-1} at step $k-1$, and θ_k denotes the weights of M at training step k. The student and teacher models are updated alternately during the training process. Figure 4.6 provides an illustration

of the self-ensembling architecture, which involves three main components: supervised learning using $\mathcal{S}$, unsupervised learning using $\mathcal{T}$, and pseudo-label learning using $\mathcal{T}$ (although not depicted in Fig. 4.6).

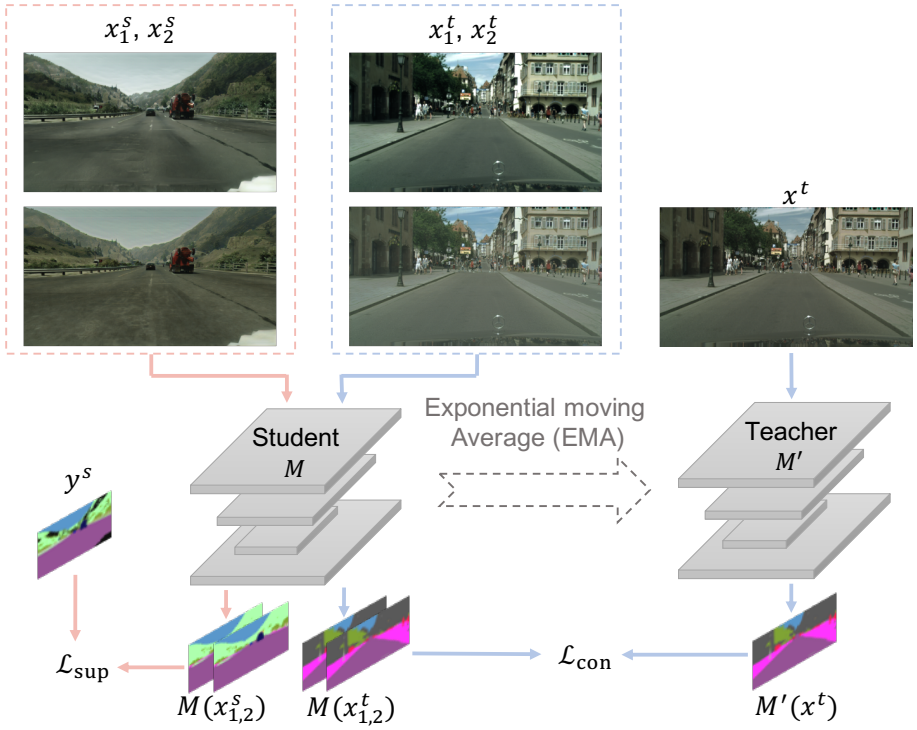


Figure 4.6: Illustration of the proposed self-ensembling method for learning intra-domain style-invariant representation. $\mathcal{L}_{\text{sup}}$ is the supervised learning loss, and $\mathcal{L}_{\text{con}}$ is the unsupervised consistency loss.

4.5.2.2 Supervised Learning with the Source Domain

Given an image-label pair $\{x^s, y^s\}$ from the source domain $\mathcal{S}$, we employ a multimodal I2I translation model to convert x^s into the target domain, resulting in two translated images x_1^s and x_2^s with distinct intra-domain styles. Subsequently, we use these translated images along with the corresponding label y^s to compute the cross-entropy loss as follows:

$$\mathcal{L}_{\text{sup}} = \mathbb{E}_{x^s,y^s}[-\frac{1}{2HW}\sum_{k}\sum_{h,w,c} y^{s(h,w,c)} \log M(x_k^s)^{(h,w,c)}], \tag{4.32}$$

where $M(x_k^s)$ represents the probability predicted by the model M for the image x_k^s. The notation (h, w, c) refers to an element in channel c at spatial position (h, w), and H and W denote the height and width of the image, respectively.

As the domain translation process is inherently imperfect, the student model acquires semantic knowledge to some extent in the target domain through the aforementioned supervised loss. Simultaneously, the learning of the intra-domain style-invariant representation is also performed in the supervised learning. Since the ground truth label is available, there is no need to impose an additional constraint, and the supervised loss inherently promotes consistent predictions for both x_1^s and x_2^s.

4.5.2.3 Unsupervised Learning with the Target Domain

Given a target-domain image x^t, we utilize the multimodal I2I translation model once more to generate two distinct copies, namely x_1^t and x_2^t, that possess diversified styles within the same domain as x^t. In other words, the translation process solely focuses on altering the intra-domain style of x^t rather than changing its domain. Since there is no available ground truth label, we employ the predictions of the teacher model M' to compute an unsupervised consistency loss as follows:

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{x^t}[\frac{1}{2}\sum_k ||M(x_k^t) - M'(x^t)||_2]. \tag{4.33}$$

The consistency loss has two components. First, it involves ensuring the consistency between the predictions of M on x_k^t and the predictions of the teacher model M' on x^t. Second, it involves maintaining consistency between the predictions of M on x_1^t and x_2^t. The first component serves as a consistency term in semi-supervised learning, which compels the student model to make predictions consistent with those of the teacher model. The teacher model, having aggregated information from multiple student models, tends to be more accurate than any individual student model. Hence, the teacher model's predictions can be used as training targets for the student model. Additionally, viewing the teacher model as an aggregation of the student models, the consistency constraint encourages the student model to be smooth in the vicinity of x^t and leads to a shift of the decision boundary towards low-density regions.

This shift enhances the model's reliability for the target domain, aligning with the "smoothness assumption" of semi-supervised learning. However, unlike traditional consistency terms in semi-supervised learning methods such as [14, 15], we sample from the vicinity of x^t by altering its intra-domain style rather than introducing trivial noise.

The second consistency component bears resemblance to the consistency between x_1^s and x_2^s in supervised learning using the source domain. Despite the absence of ground truth supervision signals in this consistency loss, both the predictions of x_1^t and x_2^t are encouraged to align with the teacher model's prediction $M'(x^t)$, thereby enforcing the learning of intra-domain style-invariant representations.

4.5.2.4 Pseudo-label Learning with the Target Domain

Similar to the methods discussed in Section 4.3 and Section 4.4, we employ pseudo-label learning once again to enhance the performance of UDA. Taking inspiration from [44], which leverages prediction uncertainty estimation to rectify pseudo-label learning, we adopt a similar rectification strategy. However, instead of using two classifiers as in [44] to estimate uncertainty, we utilize the style-diversified images x_1^t and x_2^t for this purpose. The pseudo-label loss is defined as follows:

$$\mathcal{L}_{\text{psl}} = \text{Exp}(-\text{KLD}(x_1^t, x_2^t))\mathcal{L}'_{\text{sup}} + \text{KLD}(x_1^t, x_2^t), \tag{4.34}$$

where $\text{KLD}(x_1^t, x_2^t)$ represents the Kullback–Leibler (KL) divergence between $M(x_1^t)$ and $M(x_2^t)$, while $\mathcal{L}'_{\text{sup}}$ corresponds to the cross-entropy loss using pseudo labels, following the same format as $\mathcal{L}_{\text{sup}}$.

In Eq. (4.34), the term $\text{KLD}(x_1^t, x_2^t)$ quantifies the inconsistency between the predictions of model M for the style-diversified copies x_1^t and x_2^t. When M produces divergent predictions for a pixel in x_1^t and x_2^t, it indicates ambiguity, and the model's reliability for that pixel becomes uncertain due to the ambiguous predictions. Consequently, the pseudo labels assigned to such pixels are noisier, prompting us to assign smaller weights to them in the form of $\text{Exp}(-\text{KLD}(x_1^t, x_2^t))$ within the cross-entropy loss $\mathcal{L}'_{\text{sup}}$ using pseudo labels. Simultaneously, to account for the training of pixels with smaller weights, we train the student model M to

minimize the KL divergence term $\mathrm{KLD}(x_1^t, x_2^t)$, ensuring consistency in predictions for x_1^t and x_2^t. Consequently, the training of model M is influenced less by unreliable pseudo labels compared to normal pseudo-label learning without rectification. The combined minimization of both $\mathcal{L}'_{\mathrm{sup}}$ and $\mathrm{KLD}(x_1^t, x_2^t)$ facilitates the learning of intra-domain style-invariant representations.

4.5.2.5 Training Procedure

The training process of the self-ensembling architecture consists of multiple steps. Initially, the model is trained without pseudo-label learning using the following loss function:

$$\mathcal{L}_{\mathrm{init}} = \mathcal{L}_{\mathrm{sup}} + \omega\lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}, \tag{4.35}$$

where λ_{con} is the loss weight, and ω is a weight that gradually increases from zero to one during the initial stage of training. Following this, pseudo labels are generated using the trained model, and the training process incorporates pseudo-label learning using the following loss function:

$$\mathcal{L}_{\mathrm{final}} = \mathcal{L}_{\mathrm{sup}} + \mathcal{L}_{\mathrm{psl}} + \omega\lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}. \tag{4.36}$$

In our method, we iteratively perform pseudo label generation and model training twice.

4.5.3 Multimodal Unpaired Image-to-image Translation

To generate style-diversified images for the intra-domain style-invariant representation learning, we employ an unpaired multimodal I2I translation model based on the MUNIT [89] architecture. While MUNIT has achieved success in domain transfer, it suffers from inconsistencies in the translation results, particularly in terms of semantic content. This inconsistency can hinder the learning process as the goal is to learn style-invariant representations for consistent semantic content. To address this issue, we adapt the MUNIT architecture by incorporating pixel-level semantic information into the translation process. This semantic information acts as additional guidance for the translation, improving the preservation of the original contents.

We will describe the MUNIT architecture and the modifications made in our semantic-aware MUNIT.

4.5.3.1 MUNIT Architecture

MUNIT is designed to learn disentangled representations that enable many-to-many mappings between two domains. It assumes that an image x can be decomposed and generated from a content latent code f_{cont} and a style latent code f_{sty}. The content space is shared across both domains, while the style space is specific to each domain. To extract the content code $f_{\text{cont}} = E_{\text{cont}}(x)$ and style code $f_{\text{sty}} = E_{\text{sty}}(x)$, two encoders, namely E_{cont} and E_{sty}, are trained for each domain. Additionally, a decoder G is trained to generate the translated image $x' = G(f_{\text{cont}}, f'_{\text{sty}})$, where f'_{sty} is sampled from a normal distribution $\mathcal{N}(0, I)$. Specifically, for the source domain, the encoders and decoder are denoted as $\{E^s_{\text{cont}}, E^s_{\text{sty}}, G^s\}$, and for the target domain, they are denoted as $\{E^t_{\text{cont}}, E^t_{\text{sty}}, G^t\}$.

The loss function of MUNIT consists of an adversarial loss and several reconstruction losses. Here, we take the $\mathcal{S}$-to-$\mathcal{T}$ translation as an example. Given a source domain image x^s, we aim to reconstruct the image using G^s based on its latent codes $f^s_{\text{cont}} = E^s_{\text{cont}}(x^s)$ and $f^s_{\text{sty}} = E^s_{\text{sty}}(x^s)$. This reconstruction loss is formulated as follows:

$$\mathcal{L}^x_{\text{recon}} = \mathbb{E}_{x^s}[\|x^s - G^s(f^s_{\text{cont}}, f^s_{\text{sty}})\|_1]. \tag{4.37}$$

Additionally, after translating x^s to the target domain as $x^s_{s2t} = G^t(f^s_{\text{cont}}, f'_{\text{sty}})$, we aim to reconstruct the latent codes by encoding x^s_{s2t} using the encoders of the target domain. This leads to the following reconstruction losses:

$$\mathcal{L}^{\text{cont}}_{\text{recon}} = \mathbb{E}_{x^s, x^s_{s2t}}[\|f^s_{\text{cont}} - E^t_{\text{cont}}(x^s_{s2t})\|_2], \tag{4.38}$$

$$\mathcal{L}^{\text{sty}}_{\text{recon}} = \mathbb{E}_{x^s, x^s_{s2t}}[\|f'_{\text{sty}} - E^t_{\text{sty}}(x^s_{s2t})\|_2]. \tag{4.39}$$

To ensure the realism of x^s_{s2t}, we incorporate an adversarial loss using a domain-specific discriminator D^t:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{x^s_{s2t}, x^t}[\log(1 - D^t(x^s_{s2t})) + \log D^t(x^t)]. \tag{4.40}$$

The final objective function for the $\mathcal{S}$-to-$\mathcal{T}$ translation can be defined as:

$$\min_{E,G} \max_{D} \mathcal{L}(E, G, D) = \mathcal{L}_{\text{adv}} + \lambda_x \mathcal{L}^{x}_{\text{recon}} + \lambda_c \mathcal{L}^{\text{cont}}_{\text{recon}} + \lambda_s \mathcal{L}^{\text{sty}}_{\text{recon}}, \tag{4.41}$$

where λ_x, λ_c and λ_s are hyper-parameters. The opposite $\mathcal{T}$-to-$\mathcal{S}$ translation is learned simultaneously using the same approach.

4.5.3.2 Our Semantic-aware MUNIT

To address the challenge of preserving the original semantic contents in MUNIT, we propose incorporating pixel-level semantic information into the translation process. Since ground truth labels are only available in the source domain, we cannot directly utilize them as semantic information. Instead, we discovered that the predictions generated by a pre-trained segmentation network, which can provide meaningful predictions for both domains, serve as a suitable substitute for the labels. Consequently, we pre-train a segmentation network using a simple UDA method [39] and utilize the network's predictions as the semantic information. While some information may be misleading due to potential inaccuracies in the predictions, the predicted probability distribution can contain valuable latent information beyond the binary representation of one-hot ground truth labels.

MUNIT employs the AdaIN [90] layer, a normalization layer with learnable parameters, to apply the style code to the decoder network and generate stylized images. In our method, both the style code and semantic information play a guiding role in the translation process. Therefore, it is logical to incorporate them together through the normalization layer. However, the AdaIN layer is designed to handle one-dimensional style codes and is not compatible with pixel-level inputs. To overcome this limitation, we replace the AdaIN layer with a spatially-adaptive instance normalization layer inspired by the work of Park et al. [91]. As illustrated in Fig. 4.7, the concatenation of the style code and semantic information is processed by convolutional layers, learning pixel-level affine parameters for the normalization operation. The spatially-adaptive instance normalization layer is used for all normalization layers in the decoders. By introducing the semantic information through the normalization layers, the trans-

lation network becomes aware of the semantic meanings associated with each pixel, enabling more accurate and appropriate image translation.

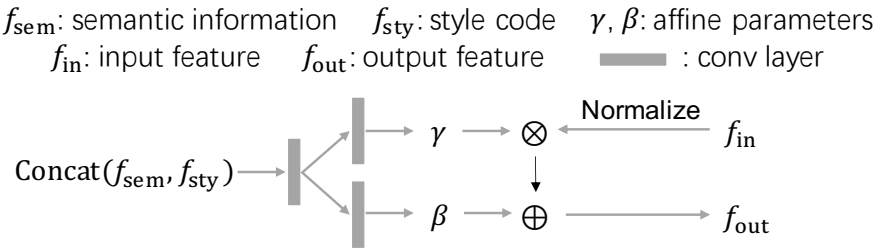


Figure 4.7: Structure of the spatially-adaptive normalization layer. Concat(f_{sem}, f_{sty}) has been resized to the spatial size of f_{in}.

The training procedure for our semantic-aware MUNIT is identical to that of MUNIT. In addition to the loss components mentioned in Section 4.5.3.1, we incorporate the cycle-consistency loss and perceptual loss, similar to MUNIT. However, we make a modification by replacing the VGG network with a segmentation network that has been pre-trained using a UDA method [39]. This pre-trained segmentation network serves the dual purpose of providing semantic information and calculating the perceptual loss.

During the training of the self-ensembling model, we dynamically generate style-diversified images, denoted as x_k^s and x_k^t (where k=1, 2), using the target-domain decoder G^t. Specifically, x_k^s represents target-domain-like images and is generated as follows:

$$x_k^s = G^t(E_{\mathrm{cont}}^s(x^s), f_k, M_{\mathrm{pre}}(x^s)), \tag{4.42}$$

where f_k represents randomly sampled style codes from the normal distribution $\mathcal{N}(0, I)$, and M_{pre} corresponds to the pre-trained segmentation model utilized for extracting semantic information. Similarly, we generate x_k^t without changing the domain as follows:

$$x_k^t = G^t(E_{\mathrm{cont}}^t(x^t), f_k, M_{\mathrm{pre}}(x^t)). \tag{4.43}$$

4.5.4 Experiments

We performed the experiments on the same benchmarks, GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes, as those in Section 4.4. Details of the datasets have been provided in Section 4.4.5.2.

4.5.4.1 Implementation Details

In our semantic-aware MUNIT, the network architectures of the encoders and decoders are identical to those of MUNIT [89], except for the normalization layers in the decoders. In the decoders, we replaced the normalization layers with spatially-adaptive instance normalization layers, which consist of three convolutional layers with 128 filters of size 3×3. For optimization, we employed the Adam optimizer with the same parameter settings as MUNIT. The hyperparameters λ_x, λ_c, and λ_s, as well as the weights assigned to the cycle-consistency loss and perceptual loss, were set to 10, 1, 1, 10, and 0.1, respectively. All input images were resized such that their longer side measures 1,024 pixels, while maintaining the original aspect ratio.

We utilized two segmentation network architectures, namely Deeplab V2 [3] with ResNet101 [1] and FCN-8s [92] with VGG16 [93]. The optimizer parameters provided by [49] were used for optimization. Both structures were trained with a batch size of 1. For the segmentation network, we set the EMA parameter α to 0.99 and the weight parameter λ_{con} to 1. The ramp-up parameter ω was defined as $\omega = \text{Exp}(-5(1-k)^2)$, where k linearly increases from zero to one during the initial 20,000 training iterations. In our experiments, we found that applying color jitter transformation as an additional intra-domain style augmenter was beneficial. Therefore, we incorporated the color jitter transformation into the generated images.

4.5.4.2 Main Results and Comparison with Results of State-of-the-art Methods

Table 4.6 and Table 4.7 present the results of our method and eight recent state-of-the-art methods. We summarize the common ideas found in the previous methods. Most methods, with the exception of PIT and PCEDA, employed pseudo labels. To address the visual domain

gap, BDL, FDA, LTIR, and PCEDA performed translation. Among these, FDA stood out by utilizing a Fourier Transform-based translation approach. Another shared component among BDL, RectPLL, SIM, and LTIR was the adoption of output-space adversarial learning. PIT distinguished itself as a unique method by exploring the domain-invariant interactive relation between image-level information and pixel-level information.

The results for the GTA5-to-Cityscapes benchmark are presented in Table 4.6. Our method demonstrated the highest performance among all the methods, as shown in Table 4.6. Specifically, our mean Intersection over Union (IoU) scores were 1.8 and 1.6 higher than the second-best methods for the two base structures, respectively. Additionally, our method outperformed all other competitive methods in 7 categories for both structure settings. For the more challenging SYNTHIA-to-Cityscapes benchmark shown in Table 4.7, mean IoUs were calculated across 13 categories and 16 categories, following the evaluation of previous studies. In the ResNet101 setting, our method achieved superior performance compared to the second-best method, with improvements of 2.2 and 1.1 in mean IoU scores for the respective category settings. In the VGG16 setting, our method exhibited a slight advantage over the second-best method, with improvements of 0.6 and 0.7 in mean IoU scores for the respective category settings. Overall, our method achieved state-of-the-art results for both benchmarks.

Table 4.6: Results of intersection over union (IoU) for GTA5-to-Cityscapes benchmark.

Method	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
Structure	ResNet101																			
BDL [49]	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	**43.6**	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
CAG-UDA [94]	90.4	51.6	83.8	34.2	27.8	**38.4**	25.3	48.4	85.4	38.2	78.1	58.6	34.6	84.7	21.9	42.7	**41.1**	29.3	37.2	50.2
RectPLL [44]	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
FDA [42]	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
SIM [95]	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
PIT [96]	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
LTIR [97]	92.9	**55.0**	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	**87.1**	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2
PCEDA [98]	91.0	49.2	85.6	37.2	29.7	33.7	38.1	39.2	85.4	35.4	85.1	61.1	32.8	84.1	45.6	46.9	0.0	34.2	44.5	50.5
Ours	**93.0**	54.0	**86.6**	**42.6**	**34.7**	35.9	40.8	43.3	**86.0**	43.2	85.4	61.5	34.4	83.7	29.2	50.1	4.0	**36.5**	**50.9**	**52.4**
BDL [49]	89.2	40.9	81.2	29.1	19.2	14.2	29.0	19.6	83.7	35.9	80.7	**54.7**	23.3	82.7	25.8	28.0	2.3	**25.7**	19.9	41.3
FDA [42]	86.1	35.1	80.6	30.8	20.4	27.5	30.0	26.0	82.1	30.3	73.6	52.5	21.7	81.7	24.0	30.5	29.9	14.6	24.0	42.2
SIM [95]	88.1	35.8	83.1	25.8	23.9	29.2	28.8	28.6	83.0	36.7	82.3	53.7	22.8	82.3	26.4	38.6	0.0	19.6	17.1	42.4
PIT [96]	86.2	35.0	82.1	31.1	22.1	23.2	29.4	28.5	79.3	31.8	81.9	52.1	23.2	80.4	**29.5**	26.9	**30.7**	20.5	1.2	41.8
LTIR [97]	92.5	**54.5**	**83.9**	34.5	25.5	**31.0**	30.4	18.0	**84.1**	39.6	83.9	53.6	19.3	81.7	21.1	13.6	17.7	12.3	6.5	42.3
PCEDA [98]	90.2	44.7	82.0	28.4	**28.4**	24.4	33.7	35.6	83.7	**40.5**	75.1	54.4	**28.2**	80.3	23.8	**39.4**	0.0	22.8	30.8	44.6
Ours	**93.7**	53.6	83.5	**35.1**	21.1	28.6	**36.2**	**42.0**	82.2	32.4	**86.5**	47.3	19.4	**83.8**	26.0	30.7	30.2	13.1	**32.1**	**46.2**

Table 4.7: Results of intersection over union (IoU) for SYNTHIA-to-Cityscapes benchmark. "mIoU" is the mean IoU over all of the 16 categories, and "mIoU*" is that over 13 categories excluding 3 categories marked by "*". Results of "-" were not reported in the papers.

Method	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	sky	person	rider	car	bus	motorcycle	bicycle	mIoU	mIoU*
Structure	ResNet101																	
BDL [49]	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	73.7	**42.2**	25.7	45.3	-	51.4
CAG-UDA [94]	84.7	40.8	81.7	7.8	0.0	35.1	13.3	22.7	**84.5**	77.6	**64.2**	27.8	80.9	19.7	22.7	48.3	44.5	52.6
RectPLL [44]	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
FDA [42]	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
SIM [95]	83.0	44.0	80.3	-	-	-	17.1	15.8	80.5	81.8	59.9	**33.1**	70.2	37.3	28.5	45.8	-	52.1
PIT [96]	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
LTIR [97]	**92.6**	53.2	79.2	-	-	-	1.6	7.5	78.6	**84.4**	52.6	20.0	82.1	34.8	14.6	39.4	-	49.3
PCEDA [98]	85.9	44.6	80.8	9.0	0.8	32.1	24.8	23.1	79.5	83.1	57.2	29.3	73.5	34.8	32.4	48.2	46.2	53.6
Ours	91.9	**54.6**	81.3	7.2	1.1	33.8	29.6	30.0	78.5	80.0	61.6	28.9	82.4	32.8	37.3	52.6	**49.0**	**57.1**
Structure	VGG16																	
BDL [49]	72.0	30.3	74.5	0.1	0.3	24.6	10.2	25.2	80.5	80.0	54.7	23.2	72.7	24.0	7.5	44.9	39.0	46.1
FDA [42]	84.2	35.1	78.0	6.1	0.4	27.0	8.5	22.1	77.2	79.6	55.5	19.9	74.8	**24.9**	14.3	40.7	40.5	47.3
PIT [96]	81.7	26.9	78.4	**6.3**	0.2	19.8	13.4	17.4	76.7	74.1	47.5	22.4	**76.0**	21.7	**19.6**	27.7	38.1	44.9
LTIR [97]	**89.8**	**48.6**	**78.9**	-	-	-	0.0	4.7	**80.6**	81.7	36.2	13.0	74.4	22.5	6.5	32.8	-	43.8
PCEDA [98]	79.7	35.2	78.7	1.4	**0.6**	23.1	10.0	**28.9**	79.6	81.2	51.2	**25.1**	72.2	24.1	16.7	**50.4**	41.1	48.7
Ours	84.6	40.3	74.5	0.5	0.1	**27.7**	**25.4**	25.1	78.0	**81.8**	**58.0**	19.4	70.5	24.3	17.7	41.5	**41.8**	**49.3**

4.5.4.3 Supplementary Results and Analyses

Ablation study. Table 4.8 presents the results of an ablation study conducted to analyze the individual contributions of each component in our method. As a baseline, we initially trained the model using only the original source-domain images without any adaptation. The baseline achieved mean IoU scores of 35.1 and 33.8 for GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes, respectively. Next, we introduced style-diversified source-domain images, without incorporating the self-ensembling architecture. This inclusion significantly improved the mean IoU scores to 46.9 and 40.5, attributed to the domain transfer and intra-domain style diversification. By leveraging the target-domain images in conjunction with the self-ensembling architecture, further enhancements were observed, resulting in mean IoU scores of 48.1 and 42.1. Lastly, the integration of pseudo-label learning provided an additional boost, elevating the mean IoU scores to 52.4 and 49.0. Overall, each component of our method contributed progressively to the improvement of the mean IoU scores for both the GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes benchmarks.

Table 4.8: Results of ablation study for the components in our method with the ResNet101 structure.

Method	Components			Mean IoU	
	$\mathcal{L}_{\text{sup}}$	$\mathcal{L}_{\text{con}}$	$\mathcal{L}_{\text{psl}}$	GTA5	SYNTHIA
Source only (non-adaptation)				35.1	33.8
Source only (style-diversified)	✓			46.9	40.5
Self-ensembling	✓	✓		48.1	42.1
Self-ensembling + pseudo-label learning	✓	✓	✓	52.4	49.0

Measures of style diversification. The intra-domain style diversification is a crucial aspect of our method, and thus we conducted a comparison of various style diversification measures, the results of which are presented in Table 4.9. Note that the results in Table 4.9 were obtained using the self-ensembling architecture without incorporating pseudo-label learning. The first measure we examined was color jitter, which involves modifying the brightness, con-

trast, saturation, and hue of an image without altering its domain. Color jitter contributed to the learning of intra-domain style-invariant representations; however, when used alone, it did not yield satisfactory results in terms of UDA performance. To improve the UDA performance, we employed CycleGAN [29], a one-to-one translation model, to translate the source-domain images to the target domain before applying color jitter. This approach resulted in improved performance for the GTA-to-Cityscapes benchmark but did not yield significant improvements for the other benchmark. Comparing these methods to our base model, MUNIT, we found that MUNIT slightly outperformed color jitter. Furthermore, applying color jitter after translation with MUNIT did not lead to further improvements. In contrast, our semantic-aware version of MUNIT, which we refer to as our I2I translation model, surpassed both the original MUNIT and all the aforementioned measures. Additionally, a slight improvement was observed when combining our I2I translation model with color jitter.

Table 4.9: Results for comparison of style diversification measures with the ResNet101 structure. "PM" denotes the measure used in the proposed method.

Style diversification measure	Mean IoU	
	GTA5	SYNTHIA
Color jitter	43.5	40.1
CycleGAN + color jitter	46.5	39.5
MUNIT	43.9	40.6
MUNIT + color jitter	43.2	40.7
Semantic-aware MUNIT	47.8	41.4
Semantic-aware MUNIT + color jitter (PM)	**48.1**	**42.1**

Number of sampled styles. To explore the potential impact of sampling more than two style-diversified copies for each image, we conducted a study on the influence of the number of sampled intra-domain styles. The objective was to determine if increasing the number of sampled styles could further enhance the performance. Surprisingly, as shown in Table 4.10, our method did not yield any significant improvement when the number of sampled styles was increased. The results indicate that sampling two styles is sufficient for achieving style-

invariant representation learning. Moreover, the lack of improvement with additional sampled styles suggests that the performance gain observed with two styles was primarily attributed to the proposed intra-domain style-invariant representation learning, rather than the presence of double copies in a mini-batch. If the number of copies in a mini-batch were the determining factor, the performance would have been expected to improve further with four or eight sampled styles. However, this was not the case, reinforcing the conclusion that the proposed intra-domain style-invariant representation learning was the key contributing factor to the observed performance improvements.

Table 4.10: Results of study on the influence of the number of sampled styles for one image with the ResNet101 structure.

Number of sampled styles	Mean IoU	
	GTA5	SYNTHIA
1	42.4	38.6
2	**48.1**	**42.1**
4	48.0	41.9
8	48.1	42.0

Comparison of translation results. Figure 4.8 presents a visual comparison of the translation results between MUNIT and our semantic-aware MUNIT. The first row showcases the translation results from GTA5 to Cityscapes, while the second row demonstrates the results of intra-domain style diversification for a target-domain image. Notably, the semantic contents in the sky region exhibit inconsistencies in MUNIT's results. However, our method significantly improves the consistency of semantic contents in the translated images, as clearly depicted in Fig. 4.8.

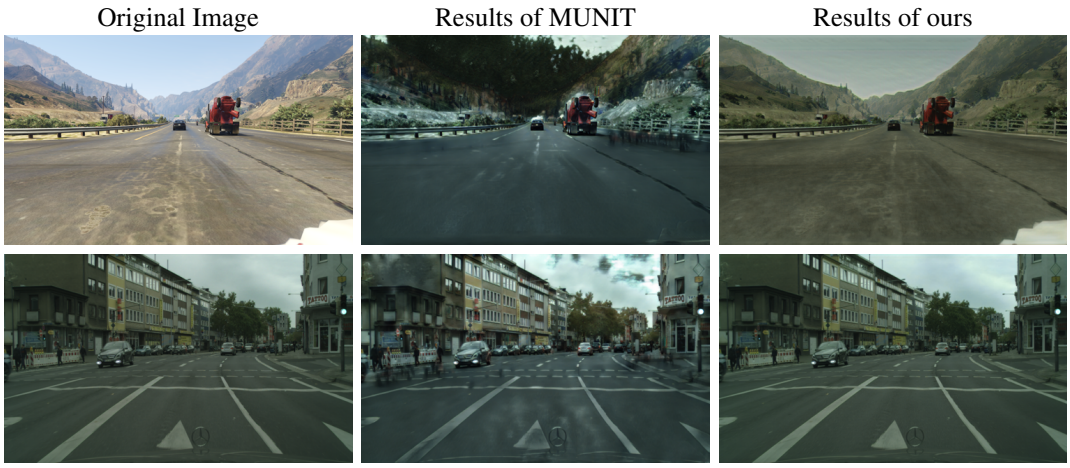


Figure 4.8: Translation results of MUNIT and the proposed semantic-aware MUNIT.

Hyper-parameter analyses. Figure 4.9 illustrates the results of our analyses for the hyper-parameters λ_{con} and α. The parameter λ_{con} represents the weight assigned to the consistency loss in the self-ensembling architecture, while α is the parameter controlling the EMA updating for the teacher model. Without pseudo-label learning, our method achieved the best performance when λ_{con} was set to 1.0 and α was set to 0.99. Examining Fig. 4.9, it is evident that using a small value of λ_{con} (e.g., 0.2) or a large value of λ_{con} (e.g., 2.0) significantly reduced the performance. On the other hand, the parameter α had only a minor impact on the performance.

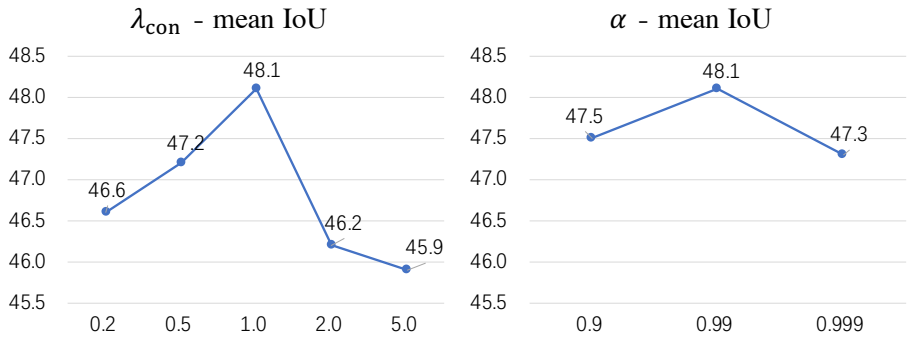


Figure 4.9: Influence of hyper-parameters λ_{con} and α on mean IoU. The experiments were conducted without pseudo-label learning on GTA5-to-Cityscapes with the ResNet101 structure.

4.5.5 Conclusion

In this section, we have presented a novel concept of learning intra-domain style-invariant representation for UDA in semantic segmentation. Based on this concept, we have developed a method that aims to enhance generalization in the target domain. Our approach involves training a semantic-aware multimodal I2I translation model to generate images with diverse intra-domain styles and consistent semantic contents. These generated images are then utilized to train the segmentation model using a self-ensembling architecture. Additionally, we incorporated pseudo-label learning to further improve the performance of our method, resulting in state-of-the-art results on two benchmark datasets. Through comprehensive experiments and analyses, we have demonstrated the effectiveness of our method in addressing UDA in semantic segmentation.

4.6 Conclusion

In this chapter, we have presented several UDA methods for object detection and semantic segmentation. Specifically, for UDA of object detection, to solve the problem of the global feature alignment that is unaware of the foreground and poorly-aligned regions, in Section 4.2, we proposed a divergence-guided feature alignment method that is aware of the information

of the important regions and leverages the information to improve the feature alignment. As to UDA of semantic segmentation, we proposed three methods to solve the problems mentioned in Section 2.5. First, in Section 4.3, for the problem of noisy pseudo labels, we proposed a method that performs symmetric domain adaptation and uses the symmetric adaptation consistency to reduce the noise in the pseudo labels. Next, in Section 4.4, for the problem of difficulty of training with the adversarial learning, we proposed a VAE-based method that trains a VAE to learn the distribution of segmentation outputs as a replacement of the adversarial learning. Finally, in Section 4.5, for the problem of neglect of the target domain's style diversity, we proposed a method with a novel concept of learning intra-domain style-invariant representation to improve the generalization in the target domain.

Chapter 5

Mitigation of Label Dependence with Model Adaptation

5.1 Introduction

The chapter focuses on model adaptation, a variant of unsupervised domain adaptation (UDA), which replaces the source-domain data with a pre-trained source-domain model and is consequently applicable to a wider range of real-world scenarios. Previous methods on model adaptation have been mainly developed for image classification which is much easier than adaptation for a semantic segmentation model. Moreover, most previous studies have been done in the single-source setting, while multi-source model adaptation (MSMA) has been rarely studied. However, the problem setting of MSMA is meaningful because pre-trained models of different domains are available in many real-world scenarios and choosing only the optimal model may be difficult and wasteful. Therefore, in this chapter, we propose the first method for MSMA of semantic segmentation, which harmonizes the different characteristics of the source domains to improve the generalization in the target domain. Moreover, we note a problem of the general multi-source setting, the too strict requirement for the label spaces to be practical in real-world applications. To improve the practicality of MSMA, we propose a new multi-source setting that relaxes the requirement for the label spaces by allowing the label spaces of the source domains to be subsets of that of the target domain. The union set of the source-domain label spaces is assumed to be equal to the target-domain label space, making

it to still be a closed-set problem. With such a relaxed multi-source setting, there may be a larger number of usable pre-trained models for MSMA in some scenarios, and the MSMA method can be applied to a wider range of applications. We name the model adaptation in the new multi-source setting as union-set multi-source model adaptation (US-MSMA). For US-MSMA of semantic segmentation, we propose a method that is on the basis of the same conception as the method for MSMA and more practical for real-world problems.

5.2 Multi-source Model Adaptation of Semantic Segmentation Learning Model-invariant Features

5.2.1 Introduction

In this section, we present an method for MSMA of semantic segmentation. Our method leverages the diverse characteristics of pre-trained models from different source domains to learn model-invariant features. The goal of model-invariant feature learning is to obtain target-domain features that have similar distributions by harmonizing the model characteristics derived from various source domains. This is achieved through latent adversarial learning between the backbone networks and classifiers of the adaptation models. By reducing domain biases and harmonizing the model characteristics, our method enhances the generalization ability of the source-domain models. The adaptation models benefit from the diverse characteristics derived from the source domains, leading to the production of more generalized features that perform well in the target domain. To create a unified model, we distill and integrate the knowledge from the adaptation models, resulting in a single model that combines the strengths of the individual models.

5.2.2 Overall Architecture

Let $\{D_i^S\}_{i=1}^k$ denote k labeled source domains, and let D^T denote the unlabeled target domain. It is assumed that $\{D_i^S\}_{i=1}^k$ and D^T share the same label set. Given unlabeled data $\{x_i^T\}_{i=1}^n$ of D^T and source-domain models $\{M_i^S\}_{i=1}^k$ pre-trained with $\{D_i^S\}_{i=1}^k$, our objective is to train a

segmentation model that can perform well in the target domain by transferring the source-domain knowledge from $\{M_i^S\}_{i=1}^k$ to the target domain.

In our method, we decompose the adaptation models $\{M_i\}_{i=1}^k$ into backbone networks $\{B_i\}_{i=1}^k$ and classifiers $\{C_i\}_{i=1}^k$, represented as $M_i(\cdot) = \sigma(C_i(B_i(\cdot)))$, where $\sigma(\cdot)$ denotes the softmax function. Our method comprises two stages. In Stage I, we perform alternate updates of $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ using an adversarial learning framework to learn the model-invariant features. In Stage II, we train a final integrated model by distilling the knowledge acquired from the adaptation models trained in Stage I. Both Stage I and Stage II build upon the baseline method with pseudo-label learning. Figure 5.1 provides an illustration of our method in the context of two source domains.

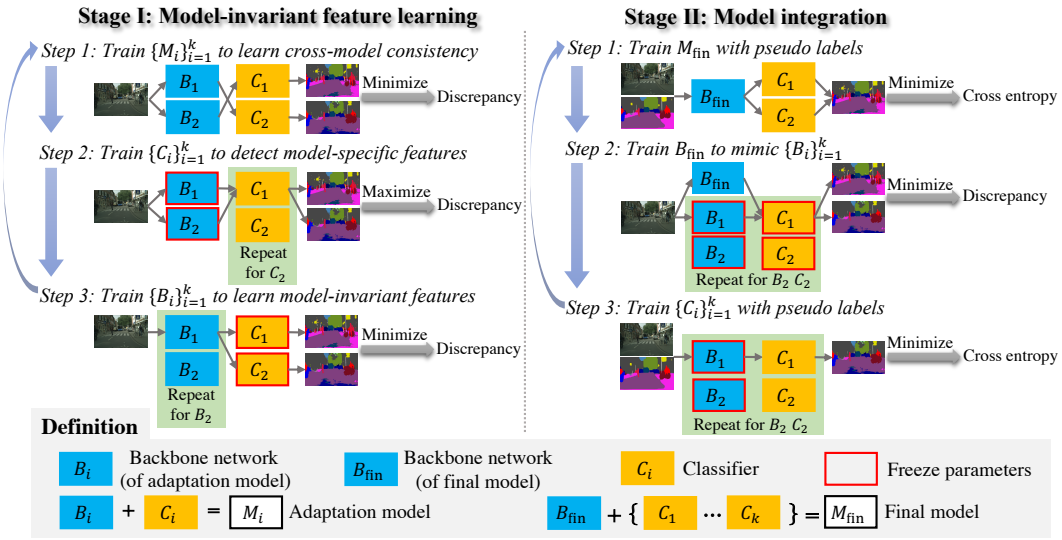


Figure 5.1: An overview of the proposed method showing the scenario of using two source-domain models. The figure excludes the depiction of pseudo-label learning in Stage I.

5.2.3 Two-stage Multi-source Modal Adaptation

5.2.3.1 Baseline with Pseudo-label Learning

Pseudo-label learning has proven to be effective in unsupervised domain adaptation [43–45, 49], and its importance is further amplified in model adaptation where supervisory signals

from the source domain are absent. Hence, we adopt pseudo-label learning as the baseline for our method. Specifically, we generate pseudo labels y^{pl} by averaging the predictions from $\{M_i^S\}_{i=1}^k$ and assign them only to pixels that are predicted to belong to a specific category with a probability higher than 0.9 or the median probability of the category. Subsequently, k models $\{M_i\}_{i=1}^k$, initialized with the pre-trained weights of $\{M_i^S\}_{i=1}^k$, are trained using the cross-entropy loss defined as follows:

$$\mathcal{L}_{\text{pl}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} -\frac{1}{HW} \sum^{H,W,C} y^{\text{pl}} \log M_i(x^T), \tag{5.1}$$

where H, W, and C denote the height, width, and number of categories, respectively, and the indices are omitted. In addition to the pseudo-label loss, which disregards pixels without a pseudo label, we incorporate the maximum squares loss [47], which maximizes the prediction confidence, into the training process:

$$\mathcal{L}_{\text{ms}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} -\frac{1}{HW} \sum^{H,W,C} M_i(x^T)^2. \tag{5.2}$$

5.2.3.2 Stage I: Model-invariant Feature Learning

The initial adaptation models $\{M_i\}_{i=1}^k$, which are initialized with the pre-trained weights of $\{M_i^S\}_{i=1}^k$, exhibit a bias towards the source domains. While learning with target-domain pseudo labels helps alleviate this bias, it is insufficient to fully address the issue. The models $\{M_i\}_{i=1}^k$ trained solely with pseudo labels still retain biases and exhibit inadequate generalization due to the presence of pseudo label noise. To mitigate the bias and enhance the generalization capabilities of $\{M_i\}_{i=1}^k$, we propose a method called model-invariant feature learning. The objective of this method is to obtain target-domain features from $\{B_i\}_{i=1}^k$ with similar distributions. As a result of the disparities between the source domains, the pre-trained models possess distinct characteristics and biases that are also inherited by $\{M_i\}_{i=1}^k$. This diversity proves advantageous for the generalization of the model-invariant features. In our method, the concept of model invariance and the similarity of feature distributions are manifested through the prediction consistency of models that combine B_i and C_j ($i \neq j$) from different adaptation models.

The model-invariant feature learning involves updating $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ in an alternating manner using adversarial learning. During this process, $\{B_i\}_{i=1}^k$ are trained to generate model-invariant features, while $\{C_i\}_{i=1}^k$ are trained to identify model-specific features from $\{B_i\}_{i=1}^k$. Specifically, to identify model-specific features from B_j, the features produced by both B_i and B_j are input into C_i, and the objective is to maximize the discrepancy between the outputs of C_i by minimizing the following loss function:

$$\mathcal{L}_{\text{cl}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{j=1, j\neq i}^{k} -\|C_i(B_i(x^T)) - C_i(B_j(x^T))\|_1. \tag{5.3}$$

As aforementioned, the model-invariant features should yield consistent predictions from C_i, regardless of whether they are generated by B_i or not. In contrast, features from B_j that result in inconsistent predictions compared to those of B_i are deemed domain-specific and can be identified by maximizing the discrepancy as Eq. (5.3). Note that the domain-specific features are not localized, but rather their domain-specific nature is assessed based on prediction consistency.

Following that, the objective of training $\{B_i\}_{i=1}^k$ is to generate model-invariant features in response to $\{C_i\}_{i=1}^k$. In typical adversarial learning manner, where the two adversaries are trained with opposing losses, $\{B_i\}_{i=1}^k$ would be updated by maximizing the loss described in Eq. (5.3). However, directly maximizing Eq. (5.3), which aims to enforce the consistency between $C_i(B_i(x^T))$ and $C_i(B_j(x^T))$, might result in excessive similarity among $\{B_i\}_{i=1}^k$. This is undesirable because if the diversity of $\{B_i\}_{i=1}^k$ is entirely eliminated, the purpose of model-invariant feature learning becomes meaningless. Consequently, instead of using the opposite loss function to Eq. (5.3), we pursue domain invariance by training B_i to minimize the discrepancy between the outputs of C_i and C_j using the following loss function:

$$\mathcal{L}_{\text{ba}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{j=1, j\neq i}^{k} \|C_i(B_i(x^T)) - C_j(B_i(x^T))\|_1. \tag{5.4}$$

The objective of minimizing the loss $\mathcal{L}_{\text{ba}}$ is to enforce domain-invariance of the features generated by B_i that result in inconsistent predictions from C_i and C_j. This adversarial learning between $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ is conducted in a latent manner, ensuring that domain-specific features are transformed into domain-invariant ones.

To further improve the domain-invariance of the features, we introduce a cross-model consistency loss, which simultaneously updates $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$. We recombine B_i and C_i from one model M_i with those from another model M_j. By minimizing the discrepancy between the outputs of the two recombined models, $C_i(B_j(\cdot))$ and $C_j(B_i(\cdot))$, the domain-invariance of the features are improved. The cross-model consistency loss is defined as follows:

$$\mathcal{L}_{\text{cm}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \|C_j(B_i(x^T)) - C_i(B_j(x^T))\|_1. \tag{5.5}$$

Additionally, minimizing the cross-model consistency loss helps to harmonize the characteristics of different adaptation models, ensuring that the trained $\{M_i\}_{i=1}^k$ exhibit similar performance.

The optimization process for the model-invariant feature learning consists of three iterative steps. In step 1, the entire models $\{M_i\}_{i=1}^k$ are updated by minimizing multiple loss functions, including the baseline losses $\mathcal{L}_{\text{pl}}$ and $\mathcal{L}_{\text{ms}}$ in Section 5.2.3.1, as well as the cross-model consistency loss $\mathcal{L}_{\text{cm}}$. In step 2, the classifiers $\{C_i\}_{i=1}^k$ are updated to minimize $\mathcal{L}_{\text{cl}}$ and also the pseudo-label loss $\mathcal{L}_{\text{pl}}$ which is introduced to prevent the degradation of performance of $\{C_i\}_{i=1}^k$. In step 3, the backbone networks $\{B_i\}_{i=1}^k$ are updated to minimize $\mathcal{L}_{\text{ba}}$.

5.2.3.3 Stage II: Model Integration

While the models $\{M_i\}_{i=1}^k$ trained with the domain-invariant feature learning have similar performance in the target domain, it is desirable to obtain a final model, denoted as M_{fin}, that surpasses all individual models $\{M_i\}_{i=1}^k$ in terms of performance. To achieve this, we introduce a model integration stage after Stage I. In this stage, we incorporate an integrated backbone network, denoted as B_{fin}, which assimilates knowledge from $\{B_i\}_{i=1}^k$. B_{fin} is combined with $\{C_i\}_{i=1}^k$ to form the final model M_{fin}, where $M_{\text{fin}}(\cdot) = \frac{1}{k}\sum_{i=1}^k \sigma(C_i(B_{\text{fin}}(\cdot)))$. Similar to $\{M_i\}_{i=1}^k$, Mfin is trained using the pseudo-label loss and the maximum squares loss, as the following equation:

$$\mathcal{L}_{\text{fin}} = \mathbb{E}_{x^T \in D^T} - \frac{1}{HW} \sum^{H,W,C} [\text{y}^{\text{pl}} \log M_{\text{fin}}(x^T) + M_{\text{fin}}(x^T)^2], \tag{5.6}$$

where y^{pl} denotes the pseudo labels, which are generated in the same way as in the baseline but with $\{M_i\}_{i=1}^k$ trained in Stage I. Additionally, B_{fin} is trained to mimic the behavior of $\{B_i\}_{i=1}^k$ with a knowledge distillation loss defined as follows:

$$\mathcal{L}_{\text{kd}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \frac{1}{HW} \sum^{H,W,C} M_i(x^T) \log \frac{M_i(x^T)}{\sigma(C_i(B_{\text{fin}}(x^T)))}. \tag{5.7}$$

During the update of B_{fin}, $\{M_i\}_{i=1}^k$ are kept frozen, and the knowledge distillation loss is employed to measure the Kullback-Leibler divergence between the predictions of M_i and the model composed of B_{fin} and C_i. Furthermore, while updating $\{C_i\}_{i=1}^k$, the pseudo-label loss $\mathcal{L}_{\text{pl}}$ (as defined in Eq. (5.1)) is used to prevent the performance degradation of $\{M_i\}_{i=1}^k$, with $\{B_i\}_{i=1}^k$ being fixed during the update. By training M_{fin} with $\mathcal{L}_{\text{fin}}$, $\mathcal{L}_{\text{kd}}$, and $\mathcal{L}_{\text{pl}}$, it can absorb and integrate the knowledge from $\{B_i\}_{i=1}^k$, ensuring that it generalizes at least as well as any individual model of $\{M_i\}_{i=1}^k$. Importantly, due to the simplicity of $\{C_i\}_{i=1}^k$, the inference time of M_{fin} is almost equivalent to that of an individual model of $\{M_i\}_{i=1}^k$.

5.2.4 Experiments

5.2.4.1 Implementation Details

In our experiments, we employed Deeplab V2 [3] with ResNet101 [1] as the segmentation network. Specifically, we used the ResNet101 backbone and the atrous spatial pyramid pooling (ASPP) classifier as the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$, respectively. The networks were trained using stochastic gradient descent (SGD) optimizer with an initial learning rate of 2.5×10^{-4}. Throughout the training process, we applied the poly policy with a power of 0.9 to decrease the learning rate. The mini-batch size was set to 1. The weights assigned to all losses were set to 1.0, except for $\mathcal{L}_{\text{ms}}$ which had a weight of 2.0.

5.2.4.2 Datasets and Adaptation Settings

For our experiments, we used the following datasets: Synscapes [77], GTA5 [82], Synthia [86] as the source domains, and Cityscapes [4] as the target domain. The source domains

(Synscapes, GTA5, and Synthia) consist of synthetic datasets that provide photo-realistic images of street scenes. On the other hand, Cityscapes is a real-world dataset containing street-scene images. The label sets for Synscapes, GTA5, and Cityscapes consist of 19 common categories, while Synthia shares a subset of 16 categories with the other datasets.

Our experiments were performed in four adaptation settings, each involving Cityscapes as the target domain. In three of these settings, we selected two out of the three source domains, while in the fourth setting, we used all three source domains. To be specific, we denoted Synscapes as S, GTA5 as G, Synthia as T, and Cityscapes as C. The adaptation settings are as follows: $S + G \rightarrow C$, $S + T \rightarrow C$, $G + T \rightarrow C$, and $S + G + T \rightarrow C$.

5.2.4.3 Methods for Comparison

Since our method is the first proposed solution for the problem of MSMA in semantic segmentation, we could only compare it to existing methods designed for related problem settings. In our evaluations, we considered two model adaptation methods: a single-source method for segmentation proposed by Fleuret et al. [61], and a multi-source method for classification proposed by Ahmed et al. [99]. Additionally, we compared our method to a UDA method proposed by Tsai et al. [39], which utilizes source-domain data during training. The single-source method [61] introduced an uncertainty reduction loss to enhance the robustness of learned feature representations, while the multi-source method [99] for classification employed an ensemble of multiple models with trainable weights. To ensure fair comparisons, we adapted these two methods and implemented them using the same loss functions $\mathcal{L}_{\text{pl}}$ and $\mathcal{L}_{\text{ms}}$ employed in our method. It's important to note that we used the same pseudo-labels, generated using multiple source-domain models, for the single-source method [61] as well. For the UDA method [39], we trained it using data from multiple source domains, treating them as a single domain during training.

5.2.4.4 Experimental Results

The experimental results of all the adaptation settings are shown in Table 5.1. The mean Intersection over Union (IoU) for the common categories shared by the source domains and the target domain is reported as the evaluation metric for each adaptation scenario. Our method incorporates a model integration stage exclusive to the full version, while the reported results for other methods represent an average result across all trained models. As demonstrated in Table 5.1, our method achieved superior performance compared to other model adaptation methods across all adaptation settings. The efficacy of each component of our method was validated through ablation studies. Notably, comparing the baseline method with only pseudo-label learning to the baseline method augmented with Stage I (model-invariant feature learning), it is evident that Stage I significantly improved adaptation performance. Including the maximum squares loss $\mathcal{L}_{\text{ms}}$ resulted in relatively smaller yet consistent enhancements in all settings when combined with the model-invariant feature learning. Furthermore, the effectiveness of Stage II (model integration) was also substantiated by comparing our method to a version without Stage II. The improvements achieved through the model integration were relatively modest, but our objective of attaining a final model surpassing the performance of all adaptation models trained in Stage I was achieved. In addition to the ablation studies, a comparison with the single-source method [61] indicated that our method is more effective for the problem of MSMA. In contrast to the multi-source method [99] employing model ensemble strategy, our method exhibited superiority not only in terms of segmentation accuracy but also inference speed, as it used only one backbone network during inference. Lastly, our method even outperformed the UDA method [39] in three settings. These results demonstrate the promising potential of using the pre-trained models as a substitute for data in the cross-domain knowledge transfer.

Table 5.1: Results of ablation studies and comparisons with other adaptation methods. "$\mathcal{L}_{\text{pl}}$", "$\mathcal{L}_{\text{ms}}$", "Stage I" and "Stage II" are the components of the proposed method. "UR" and "ME" indicate the uncertainty reduction and the model ensemble proposed by the other methods.

Method	Components						Mean IoUs in different settings			
	$\mathcal{L}_{\text{pl}}$	$\mathcal{L}_{\text{ms}}$	Stage I	Stage II	UR	ME	$S+G$	$S+T$	$G+T$	$S+G+T$
Baseline	✓						47.5	50.1	47.0	52.4
+ $\mathcal{L}_{\text{ms}}$	✓	✓					50.1	52.4	49.0	54.4
+ Stage I	✓		✓				50.4	52.3	49.0	54.2
+ $\mathcal{L}_{\text{ms}}$ + Stage I	✓	✓	✓				52.0	53.2	49.9	55.0
+ $\mathcal{L}_{\text{ms}}$ + Stage I + Stage II (**our method**)	✓	✓	✓	✓			**52.4**	53.6	**50.5**	**55.2**
Uncertainty reduction [61]	✓	✓			✓		49.9	52.5	49.1	54.4
Weighted combination [99]	✓	✓				✓	51.9	52.1	47.8	54.3
Unsupervised domain adaptation [39]							51.6	**54.0**	47.8	53.7

5.2.5 Conclusion

In this section, we have presented an innovative method for solving the problem of MSMA in semantic segmentation which exhibits promising potential for practical applications. In comparison to previous model adaptation methods, our method enhances adaptation performance by using multiple source-domain models to acquire model-invariant features that possess greater generalizability to the target domain. Furthermore, we incorporated a model integration stage to obtain a final model that outperforms all the adapted models. Through experiments across four adaptation settings, the effectiveness and superiority of our method have been validated.

5.3 Union-set Multi-source Model Adaptation of Semantic Segmentation Learning Model-invariant Features

5.3.1 Introduction

In this section, to improve the practicality of MSMA, we propose a generalized version of MSMA called union-set multi-source model adaptation (US-MSMA). Specifically, we modify the multi-source setting in Section 5.2 with a relaxation of the requirements for the label spaces. In our US-MSMA setting, the requirement is for the union of all the source-domain label spaces, rather than the individual label spaces of each source domain, to be equal to the label space of the target domain. This implies that the label space of each source domain is expected to be a subset of the target-domain label space, but not necessarily identical to it. This relaxation significantly expands the applicability of MSMA, making it more versatile. Additionally, it enables the selection of source domains from a larger pool of candidates, which can potentially enhance adaptation performance. For instance, a high-performance model trained in a source domain with high-quality labels for a subset of the target-domain classes can be incorporated into the US-MSMA training process, thereby improving adaptation performance specifically for those classes. The generalized multi-source setting aligns particularly well with model adaptation, thanks to the cost-effectiveness associated with integrating pre-trained

models.

For US-MSMA of semantic segmentation, we propose a two-stage approach that shares similarities with the MSMA method discussed in Section 5.2. Our method, like the one for MSMA, comprises a model adaptation stage and a model integration stage. During the model adaptation stage, we focus on learning model-invariant features to align the diverse model characteristics originating from various source domains. Subsequently, in the model integration stage, we leverage the knowledge distilled from the adapted models to train a final model.

5.3.2 Overall Architecture

Let $\{D_i^S\}_{i=1}^k$ represent a collection of k labeled source domains with class sets $\{\Phi_i^S\}_{i=1}^k$, while D^T denotes the unlabeled target domain with its own class set Φ^T. Unlike the general multi-source setting where each of $\{\Phi_i^S\}_{i=1}^k$ must strictly match Φ^T, our US-MSMA setting allows for the assumption that the union of $\{\Phi_i^S\}_{i=1}^k$ is equivalent to Φ^T, expressed as $\Phi_1^S \bigcup \Phi_2^S \cdots \bigcup \Phi_k^S = \Phi^T$. With access to unlabeled data $\{x_i^T\}_{i=1}^n$ from D^T and k pre-trained models $\{M_i\}_{i=1}^k$ trained on $\{D_i^S\}_{i=1}^k$ respectively, our objective is to develop a model capable of transferring knowledge from $\{D_i^S\}_{i=1}^k$ to D^T and consequently achieve satisfactory performance in D^T.

Figure 5.2 provides an overview of the proposed method, which encompasses two stages: the model adaptation stage and the model integration stage. In Stage I, we perform the model adaptation process by retraining the pre-trained source-domain models $\{M_i\}_{i=1}^k$ using the target domain D^T. To transfer source-domain knowledge to the target domain, we leverage self-training techniques (not depicted in Fig. 5.2) by training $\{M_i\}_{i=1}^k$ with pseudo labels assigned to the samples from D^T. Additionally, we enhance the adaptation process through model-invariant feature learning. Moving on to Stage II, we proceed to train a final model, denoted as M_{fin}, by distilling and integrating the knowledge acquired from $\{M_i\}_{i=1}^k$ trained in Stage I. As shown in Fig. 5.2, $\{M_i\}_{i=1}^k$ represents individual models composed of a backbone B_i and a classifier C_i, while M_{fin} corresponds to an ensemble model consisting of an integration backbone B_{fin} and all the classifiers $\{C_i\}_{i=1}^k$. The classifiers of the ensemble model are combined using a classifier ensemble strategy.

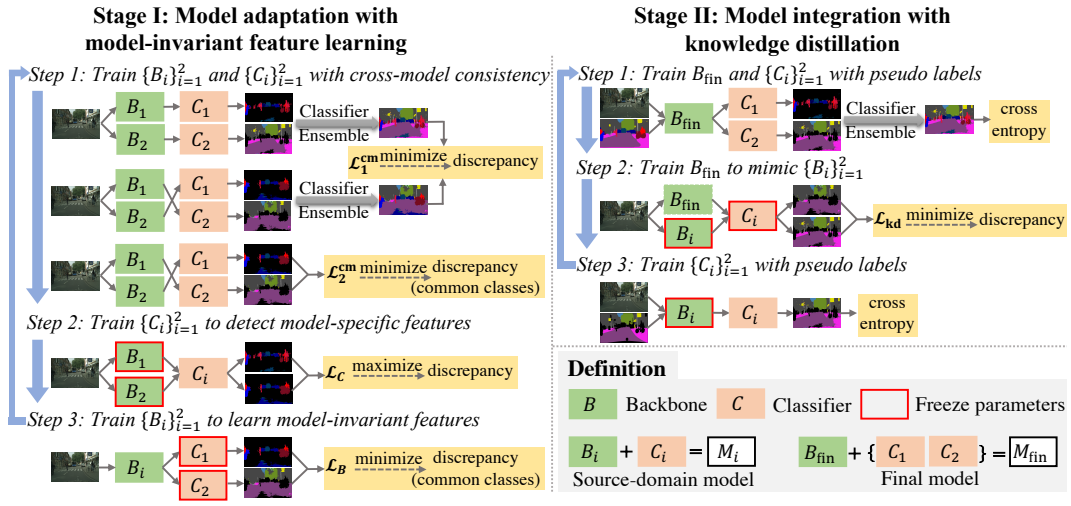


Figure 5.2: An overview of the proposed method. For the ease of understanding, we show the case of using two source-domain models. The pseudo-label learning of Stage I are omitted in the figure.

During both I and II, we make predictions on the probability distribution across all target-domain classes, denoted as Φ^T, using the classifiers $\{C_i\}_{i=1}^k$. However, since the source-domain class sets $\{\Phi_i^S\}_{i=1}^k$ may not be equivalent to Φ^T, it is possible that individual C_i may not provide predictions for the entire Φ^T. Additionally, due to the potential variation in class sets, a simple averaging of predictions from $\{C_i\}_{i=1}^k$ is not feasible. Therefore, we employ a classifier ensemble strategy to obtain complete predictions by simultaneously combining and averaging the outputs from $\{C_i\}_{i=1}^k$. Specifically, we compute the unnormalized logits of the complete prediction by averaging the logits of each class over $\{C_i\}_{i=1}^k$ using the following equation:

$$l_c(\cdot) = \frac{1}{\sum_{i=1}^k \mathbb{1}(c \in \Phi_i^S)} \sum_{i=1}^k C_{i,c}(\cdot), \ \forall c \in \Phi^T, \tag{5.8}$$

where $\mathbb{1}(\cdot)$ represents the indicator function, and $C_{i,c}(\cdot)$ denotes the logits of class c predicted by C_i if $c \in \Phi_i^S$, otherwise it is considered as zero. The calculated logits are subsequently normalized using the Softmax function to obtain the predicted probability distribution within the target-domain label space. This approach allows us to obtain predictions within the target-domain label space by using classifiers $\{C_i\}_{i=1}^k$ with incomplete class sets, without the need

to train a new classifier. The classifier ensemble operation is referred to as En(·) from here onwards.

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

In Stage I, our model adaptation process builds upon pseudo-label learning and incorporates model-invariant feature learning to enhance the adaptation performance. The objective of model-invariant feature learning is to mitigate the domain biases inherent in $\{M_i\}_{i=1}^k$. As the pre-trained $\{M_i\}_{i=1}^k$ exhibit diverse characteristics stemming from the source domains, we aim to reduce the domain bias of each M_i by aligning the characteristics of $\{M_i\}_{i=1}^k$. To achieve this, we train the backbones $\{B_i\}_{i=1}^k$ to generate features with similar distributions, referred to as model-invariant features. This learning process involves three iterative steps: the initial step for cross-model consistency and subsequent two steps for adversarial learning between the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$, as depicted on the left side of Fig. 5.2. We elaborate on each component of I as follows.

5.3.3.1 Pseudo-label Learning

To transfer the knowledge from the source domains to the target domain, we employ pseudo-label learning for both individual models $\{M_i\}_{i=1}^k$ and ensemble models consisting of a backbone B_i $(i = 1, \ldots, k)$ and all the classifiers $\{C_i\}_{i=1}^k$. For this purpose, we utilize the pre-trained $\{M_i\}_{i=1}^k$ to generate pseudo labels $\{y_i\}_{i=1}^k$ in the source-domain label spaces, as well as pseudo labels y^T in the target-domain label space for each image in D^T. To generate pseudo labels in the target-domain label space, we combine and average the predictions of $\{M_i\}_{i=1}^k$, but with a different approach than the classifier ensemble method described earlier, as $\{M_i\}_{i=1}^k$ are independently pre-trained. Specifically, we first assign 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. Here, $M_{i,c}(\cdot)$ represents the probability of class c predicted by M_i, and $M_{i,0}(\cdot)$ denotes the probability of the other classes not in Φ_i^S. We then average these probability distributions $\{p_i\}_{i=1}^k$ and assign pseudo labels based on the average

prediction. In the process of pseudo-label learning, we employ the cross-entropy loss function $\mathrm{CE}(logits, target)$ to train $\{M_i\}_{i=1}^k$, as defined in the following equation:

$$\mathcal{L}_{\mathrm{pl}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} [\mathrm{CE}(C_i(B_i(x^T)), y_i) + \mathrm{CE}(\mathrm{En}(\{C_j(B_i(x^T))\}_{j=1}^k), y_T)], \tag{5.9}$$

where $\mathrm{En}(\cdot)$ denotes the classifier ensemble operation discussed in Section 5.3.2.

5.3.3.2 Cross-model Consistency

Based on the assumption that a backbone capable of producing model-invariant features should be compatible with any classifier, we introduce a random recombination process for $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ to create k new models $\{C_{\mathrm{ma}(i)}(B_i(\cdot))\}_{i=1}^k$, where $\mathrm{ma}(i)$ represents the index of the classifier matched with B_i. These recombined models are trained based on cross-model consistency, which involves both overall consistency of ensemble predictions and per-class consistency of logits generated by individual classifiers. For overall consistency, we apply the classifier ensemble operation to predictions from 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 as follows:

$$\mathcal{L}_1^{\mathrm{cm}} = \mathbb{E}_{x^T \in D^T} \|\sigma(\mathrm{En}(\{C_i(B_i(x^T))\}_{i=1}^k)) - \sigma(\mathrm{En}(\{C_{\mathrm{ma}(i)}(B_i(x^T))\}_{i=1}^k))\|_1, \tag{5.10}$$

where $\sigma(\cdot)$ denotes the Softmax function. For per-class consistency, we train the recombined models to produce logits that are consistent for each class. We compute the average logits for each class and minimize the discrepancy between the output logits and the average logits using the following equation:

$$\mathcal{L}_2^{\mathrm{cm}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{c}^{\Phi_i^S} \|C_{\mathrm{ma}(i),c}(B_i(x^T)) - \delta_c(x^T)\|_1, \tag{5.11}$$

where $\delta_c(\cdot)$ represents the average logits for class c, calculated as:

$$\delta_c(\cdot) = \frac{1}{\sum_{i=1}^k \mathbb{1}(c \in \Phi_i^S)} \sum_{i=1}^{k} C_{\mathrm{ma}(i),c}(B_i(\cdot)), \tag{5.12}$$

where $C_{\text{ma}(i),c}(\cdot)$ represents the logits for class c produced by $C_{\text{ma}(i)}$ if $c \in \Phi^S_{\text{ma}(i)}$, and zero otherwise. By recombining and training the models using $\mathcal{L}_1^{\text{cm}}$ and $\mathcal{L}_2^{\text{cm}}$ for overall and per-class consistency, respectively, the features generated by the backbones are enforced to have similar distributions, thereby achieving model-invariant characteristics.

5.3.3.3 Adversarial Learning

To further enhance the learning of model-invariant features, we introduce adversarial learning between the backbones $\{B_i\}_{i=1}^k$ and the classifiers $\{C_i\}_{i=1}^k$, in addition to the cross-model consistency. This adversarial learning process 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. These steps are performed iteratively, with the parameters of $\{B_i\}_{i=1}^k$ ($\{C_i\}_{i=1}^k$) being frozen while updating the parameters of $\{C_i\}_{i=1}^k$ ($\{B_i\}_{i=1}^k$).

To train C_i ($i = 1, \ldots, k$), we input features from B_i and B_j ($j \neq i$) into C_i and aim to maximize the discrepancy between their predictions by minimizing the following loss function:

$$\mathcal{L}_C = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} [\sum_{j=1}^{k} -\|C_i(B_i(x^T)) - C_i(B_j(x^T))\|_1 + \text{CE}(C_i(B_i(x^T)), y_i)], \tag{5.13}$$

where we include a cross-entropy term to prevent the degradation of the recognition ability of $\{C_i\}_{i=1}^k$ while maximizing the discrepancy. By updating C_i using $\mathcal{L}_C$, we aim to detect domain-specific features from B_j that lead to inconsistent predictions compared to those obtained using the features from B_i.

We train $\{B_i\}_{i=1}^k$ using a loss function that calculates the discrepancy between the output logits and the average logits, considering features from each of the backbones separately. The loss function is defined as follows:

$$\mathcal{L}_B = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \sum_{j=1}^{k} [\sum_{c}^{\Phi_j^S} \|C_{j,c}(B_i(x^T)) - \delta_{i,c}(x^T)\|_1 + \text{CE}(C_j(B_i(x^T)), y_j)], \tag{5.14}$$

where $\delta_{i,c}(\cdot)$ represents the average logits of class c using the features from B_i. It is computed as:

$$\delta_{i,c}(\cdot) = \frac{1}{\sum_{j=1}^{k} \mathbb{1}(c \in \Phi_j^S)} \sum_{j=1}^{k} C_{j,c}(B_i(\cdot)). \tag{5.15}$$

By minimizing the L1-norm term of $\mathcal{L}_B$, each backbone is trained to produce features that yield per-class consistent logits across different classifiers, thereby promoting domain invariance. However, since the L1-norm term only considers classes shared by multiple classifiers, we include an additional cross-entropy term to ensure that each backbone is compatible with all the classifiers during training.

In contrast to the typical adversarial learning, our method involves adversarial learning between two groups of backbones $\{B_i\}_{i=1}^k$ and classifiers $\{C_i\}_{i=1}^k$, rather than specific pairwise opponents. Additionally, we employ a distinct loss function for training $\{B_i\}_{i=1}^k$, which differs from the loss used for $\{C_i\}_{i=1}^k$. This divergence prevents excessive similarity among $\{B_i\}_{i=1}^k$ by avoiding the direct minimization of the term $\|C_i(B_i(x^T)) - C_i(B_j(x^T))\|_1$ in $\mathcal{L}_C$. Our adversarial learning method is designed to be compatible with cross-model consistency and further strengthens the model-invariance of the learned features.

The training process of Stage I consists of three steps, which are repeated iteratively. In Step 1, both $\{B_i\}_{i=1}^k$ and $\{C_i\}_{i=1}^k$ are updated simultaneously using the loss $\mathcal{L}_{\text{pl}}$ from pseudo-label learning, as well as the losses $\mathcal{L}_1^{\text{cm}}$ and $\mathcal{L}^{\text{cm}}2$ from cross-model consistency. Subsequently, Step 2 updates $\{C_i\}_{i=1}^k$ using the loss function $\mathcal{L}_C$, while Step 3 updates $\{B_i\}_{i=1}^k$ using the loss function $\mathcal{L}_B$.

5.3.4 Stage II: Model Integration with Knowledge Distillation

To achieve optimal performance without relying on any individual model, we introduce the model integration stage, where the knowledge from $\{M_i\}_{i=1}^k$ is distilled and used to train a final model M_{fin}. This final model consists of an integration backbone B_{fin} along with all the classifiers $\{C_i\}_{i=1}^k$. The ensemble prediction, obtained by aggregating the predictions from $\{C_i(B_{\text{fin}}(\cdot))\}_{i=1}^k$, serves as the final prediction. During this stage, the parameters of the backbones $\{B_i\}_{i=1}^k$ remain frozen.

Similar to Stage I, the final model M_{fin} undergoes training using a loss function that update both B_{fin} and $\{C_i\}_{i=1}^k$, as well as separate losses that update B_{fin} and $\{C_i\}_{i=1}^k$ individually. Firstly, B_{fin} and $\{C_i\}_{i=1}^k$ are jointly trained by minimizing the cross-entropy loss $\text{CE}(\text{En}(\{C_i(B_{\text{fin}}(\cdot))\}_{i=1}^k), y_T)$

for the ensemble prediction using pseudo labels generated by $\{M_i\}_{i=1}^k$ trained in Stage I. Following this, B_{fin} is trained to mimic the behavior of $\{B_i\}_{i=1}^k$ using a knowledge distillation loss defined as:

$$\mathcal{L}_{\text{kd}} = \mathbb{E}_{x^T \in D^T} \sum_{i=1}^{k} \text{KLD}(C_i(B_{\text{fin}}(x^T)), C_i(B_i(x^T))), \tag{5.16}$$

where KLD($input, target$) represents the Kullback–Leibler divergence. Additionally, to maintain compatibility among $\{B_i, C_i\}_{i=1}^k$, $\{C_i\}_{i=1}^k$ is trained by minimizing $\sum_{i=1}^k \text{CE}(C_i(B_i(x^T)), y_i)$ for individual classifier predictions. Through training with these aforementioned losses, M_{fin} assimilates the knowledge acquired from $\{M_i\}_{i=1}^k$ in Stage I, resulting in superior performance compared to any individual model of $\{M_i\}_{i=1}^k$. Furthermore, due to the lightweight nature of $\{C_i\}_{i=1}^k$, the inference speed of M_{fin} is nearly equivalent to that of an individual model of $\{M_i\}_{i=1}^k$.

5.3.5 Experiments

5.3.5.1 Implementation Details

The segmentation network and training parameters are the same as those used in the method proposed in Section 5.2. All the losses were assigned equal weights of 1.0. Additionally, we incorporated a maximum squares loss [47] as an additional component to complement the losses $\mathcal{L}_{\text{pl}}$ and $\mathcal{L}_B$ in Stage I, as well as the loss $\text{CE}(\text{En}(\{C_i(B_{\text{fin}}(\cdot))\}_{i=1}^k), y_T)$ in Stage II, with the aim of reducing prediction uncertainty.

5.3.5.2 Datasets and Adaptation Settings

We used multiple datasets for our experiments: Synscapes dataset [77], GTA5 dataset [82], Synthia dataset [86], and Cityscapes dataset [4]. Among these, Synscapes, GTA5, and Synthia serve as the source domains, while Cityscapes serves as the target domain. The source domains consist of synthetic datasets with photo-realistic street scene images, while Cityscapes is a real-world dataset comprising street-scene images. The label space is shared between Synscapes, GTA5, and Cityscapes, containing 19 classes. Synthia, on the other hand, shares a subset of

16 classes with the other datasets. Henceforth, we will refer to Synscapes, GTA5, and Synthia as S, G, and T respectively.

We conducted comprehensive experiments by considering all possible combinations of the source domains: $S+G$, $S+T$, $G+T$, and $S+G+T$. Additionally, our method is designed to accommodate various label space configurations in the source domains, we therefore evaluated our method in three distinct label space settings: non-overlapping, partly-overlapping, and fully-overlapping. In the non-overlapping setting, the target-domain classes were divided into subsets with no classes in common, and these subsets were assigned to the respective source domains. In the partly-overlapping setting, the background classes were shared among the domains, while the foreground classes were split between the source domains. Detailed information regarding the class distributions for the non-overlapping and partly-overlapping settings can be found in Table 5.2. In the fully-overlapping setting, most of the classes were shared across all domains, with the exception of three classes (terrain, truck, train) that were absent in Synthia. However, we still included these classes in the experiments with $S+T$ and $G+T$.

Table 5.2: Class distributions of the non-overlapping setting and the partly-overlapping setting.

Setting			Source domain	bicycle	motorcycle	train	bus	truck	car	rider	person	sky	terrain	vegetation	t-sign	t-light	pole	fence	wall	building	sidewalk	road
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	✓	✓					✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓

5.3.5.3 Methods for Comparison

Given the absence of existing methods specifically designed for our problem setting, we made slight modifications to three methods proposed for related problem settings to conduct fair comparisons. These methods include a UDA method [39] for semantic segmentation, a single-source model adaptation (SSMA) method [61] for semantic segmentation, and an MSMA method [99] for image classification. For the UDA method based on adversarial learning, we employed domain-specific discriminators and classifiers for the source domains and

trained a shared backbone. In both the UDA method and the SSMA method, we obtained complete predictions by first aligning the predictions in the source-domain label spaces to the target-domain label space and then averaging them, following the same procedure used to generate pseudo labels in our method. As for the MSMA method, which learns a set of weights for combining models, we applied the same procedure of aligning each model's prediction to the target-domain label space and calculated a weighted average prediction using the learned weights. To facilitate direct comparisons with the SSMA method and the MSMA method, we trained these methods using the same maximum squares loss and cross-entropy loss with identical pseudo labels to those employed in our method.

In the fully-overlapping setting, there are slight differences in the implementations compared to the above description. Since the source domains and the target domain share the same label space, we made some adjustments. For the SSMA method, instead of performing model ensemble, we evaluated the independent performances of all the trained models and averaged them to obtain the final performance. Regarding the MSMA method, there were no changes in its evaluation since the weighted model ensemble is the core aspect of this method. As for the UDA method, we used domain-specific discriminators as before, but only one classifier was employed in this case.

5.3.5.4 Results in Non-overlapping Setting

The results in the non-overlapping setting are presented in Table 5.3. The evaluation metric used was the mean Intersection over Union (IoU) across all the target-domain classes. In the case of our incomplete method versions, where multiple models were trained without a final model, we evaluated the ensemble models comprising a single backbone and multiple classifiers independently and reported their average performance. As shown in Table 5.3, the performance consistently improved with the inclusion of two key components: cross-model consistency and adversarial learning, both of which contribute to model-invariant feature learning. The introduction of the maximum squares loss in Stage I of our method further enhanced the results. Finally, through the model integration in Stage II, we achieved the best performance,

showing significant improvements compared to the baseline that relied solely on pseudo-label learning. These results from the ablation study demonstrate the effectiveness of each component in our method.

In the non-overlapping setting, our method achieved superior performance compared to other adaptation methods, even with only Stage I. Furthermore, both the SSMA and MSMA methods require inference with multiple models to obtain complete predictions, whereas our method's final model consists of only one backbone, resulting in significantly faster inference times. The MSMA method failed in the non-overlapping setting because the weighted model ensemble method is rendered meaningless without any shared classes.

Qualitative results are shown in Fig. 5.3, which displays two examples comparing our method to source-domain models without adaptation using the source domains of $S+G$. The figure illustrates that our method improved the segmentation of both background and foreground classes. In the upper example, the predictions for sky, sidewalk, and vegetation showed notable improvement after adaptation. Similarly, in the lower example, classes such as sidewalk, traffic sign, vegetation, and rider were segmented more accurately in the results obtained using our method.

Table 5.3: Results in the non-overlapping setting. PSL: pseudo-label learning. CMC: cross-model consistency. ADV: adversarial learning. MSL: maximum squares loss. MI: model integration. PM: proposed method.

Method	$S+G$	$S+T$	$G+T$	$S+G+T$
PSL	42.3	38.8	35.8	40.4
PSL+CMC	43.2	39.1	36.1	40.6
PSL+CMC+ADV	44.0	39.8	36.4	41.5
PSL+CMC+ADV+MSL (**=Stage I of PM**)	45.8	41.4	37.2	42.1
PSL+CMC+ADV+MSL+MI (**=PM**)	**46.6**	**42.3**	**37.9**	**44.2**
SSMA [61]	43.5	40.6	37.0	41.9
MSMA [99]	30.2	26.0	20.7	22.7
UDA [39]	45.7	39.2	35.9	41.1

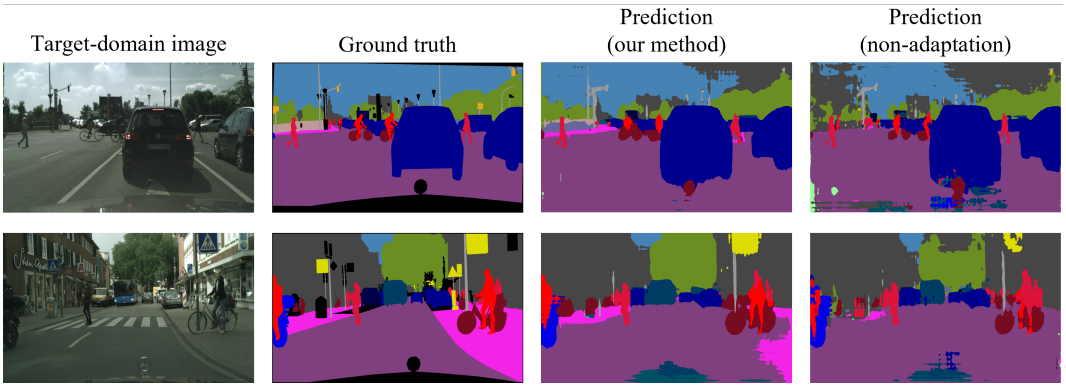


Figure 5.3: Examples of the qualitative results of the proposed method and the source-domain models without adaptation in the non-overlapping setting of $S+G$.

5.3.5.5 Results in Partly-overlapping Setting

In the partly-overlapping setting, we conducted experiments identical to those in the non-overlapping setting, except for the inclusion of $S+G+T$. The results are presented in Table 5.4. Similar to the findings in the non-overlapping setting's ablation study, each component consistently contributed to performance improvement. When compared to other adaptation methods, our method once again achieved the best overall performance, as indicated in Table 5.4. However, unlike the non-overlapping setting, the version of our method with only Stage I did not surpass the SSMA and UDA methods in performance. The presence of common classes in the partly-overlapping setting provided significant benefits to the SSMA method through model ensembling for obtaining complete predictions. However, this came at the cost of decreased inference speed by several times. Similarly, with the existence of common classes, the MSMA method achieved reasonable performance utilizing weighted model ensembling. The UDA method closely approached our method's performance in the $S+G$ and $S+T$ settings, but it required access to the source-domain data.

Table 5.4: Results in the partly-overlapping setting. PSL: pseudo-label learning. CMC: cross-model consistency. ADV: adversarial learning. MSL: maximum squares loss. MI: model integration. PM: proposed method.

Method	$S+G$	$S+T$	$G+T$
PSL	44.2	44.0	39.1
PSL+CMC	46.0	44.8	39.6
PSL+CMC+ADV	46.6	45.2	40.6
PSL+CMC+ADV+MSL (**=Stage I of PM**)	47.4	45.9	42.2
PSL+CMC+ADV+MSL+MI (**=PM**)	**48.3**	**47.2**	**43.5**
SSMA [61]	47.2	46.3	42.7
MSMA [99]	46.5	44.1	41.9
UDA [39]	47.9	46.9	41.6

5.3.5.6 Results in Fully-overlapping Setting

We also conducted experiments in the fully-overlapping setting, which represents the general multi-source model adaptation scenario. However, we did not perform experiments in the $S+G+T$ setting since introducing T did not lead to any improvements compared to the performance in $S+G$ due to the significantly larger domain gap between T and the target domain. The results in the fully-overlapping setting are presented in Table 5.5. Similar to the other two settings, the ablation study results confirmed the effectiveness of each component of our method. However, the maximum squares loss had minimal contributions in the fully-overlapping setting. This can be attributed to the more accurate pseudo labels generated in this setting, which reduced the significance of the maximum squares loss. When comparing our method to other adaptation methods, our method consistently outperformed the SSMA method. However, in the $S+T$ setting, the UDA method achieved slightly better performance than ours. Additionally, the MSMA method attained the same performance as our method in $S+G$, using two source domains with closer domain gaps to the target domain. This indicated that the efficiency of the weighted model ensemble is maximized when the source domains share the same label space and exhibit similar domain gaps with the target domain. Overall, our method provides the best cost-performance ratio, considering the inference speed and the

requirement for access to source-domain data.

Table 5.5: Results in the fully-overlapping setting. PSL: pseudo-label learning. CMC: cross-model consistency. ADV: adversarial learning. MSL: maximum squares loss. MI: model integration. PM: proposed method.

Method	*S*+*G*	*S*+*T*	*G*+*T*
PSL	47.6	45.7	44.0
PSL+CMC	49.4	46.6	44.7
PSL+CMC+ADV	50.9	47.7	45.7
PSL+CMC+ADV+MSL (**=Stage I of PM**)	51.0	47.8	45.7
PSL+CMC+ADV+MSL+MI (**=PM**)	**51.7**	48.7	**46.2**
SSMA [61]	49.8	47.6	45.5
MSMA [99]	**51.7**	47.2	44.2
UDA [39]	51.6	**49.0**	45.8

5.3.6 Conclusion

In this section, we have presented a novel problem called union-set multi-source model adaptation, which offers broader applicability to practical scenarios compared to the general multi-source setting by requiring the union of the source-domain label spaces to match the target-domain label space. To tackle the problem of union-set multi-source model adaptation for semantic segmentation, we proposed a method based on a novel learning strategy called model-invariant feature learning. This strategy aims to enhance generalization in the target domain by harmonizing the diverse characteristics of the source-domain models. Additionally, we incorporated a model integration stage, which distills knowledge from the adapted models and trains a final model with improved performance. Through comprehensive ablation studies, we validated the effectiveness of each component of our method. Furthermore, extensive experiments conducted in various settings demonstrated the superiority of our method over previous adaptation methods.

5.4 Conclusion

In this chapter, we have presented two methods for model adaptation of semantic segmentation in two different multi-source setting: a general multi-source setting where all the source-domain label spaces are required to be equal to the target-domain label space, and a union-set multi-source setting where only the union set of the source-domain label spaces is required to be equal to the target-domain label space. The union-set multi-source setting, which is first proposed in our study, is more practical in real-world applications than the general multi-source setting. The methods for the two settings are developed on the basis of the same idea, model-invariant feature learning, which harmonizes the model characteristics derived from different source domains to produce more generalizable features thereby performing better in the target domain. Experimental results demonstrated the effectiveness of the proposed methods and the superiority to the previous methods.

Chapter 6

Conclusion

As a conclusion of this thesis, this chapter summarizes the proposition of this thesis and shows future directions.

6.1 Summary of Proposition in this Thesis

In this thesis, we focused a common problem in applications of image recognition technologies, the dependence on labeled data. To mitigate the label dependence thereby improving the practicality, we conducted studies in three directions: semi-supervised learning, unsupervised domain adaptation, and model adaptation.

In Chapter 3, in the direction of semi-supervised learning, we proposed a tri-training based semi-supervised learning method for chronic gastritis classification using gastric X-ray images. The method uses the tri-training architecture to improve the pseudo label generation and can achieve high performance for gastritis diagnosis even with a small amount of labeled data.

In Chapter 4, in the direction of unsupervised domain adaptation, we proposed several methods for the two challenging and important tasks, object detection and semantic segmentation. For unsupervised domain adaptation of object detection, we proposed a divergence-guided feature alignment method. By introducing the divergence-based guidance mechanism, the feature alignment is aware of foreground regions and poorly-aligned regions and can thus improve the adaptation performance even more. For unsupervised domain adaptation of semantic segmentation, we proposed three methods including a symmetric adaptation-based method, a

variational autoencoder-based method, and a method based on the intra-domain style-invariant representation learning. The first method performs symmetric domain adaptation via adversarial learning and uses the symmetric adaptation consistency to reduce the noise in the pseudo labels. The second method uses a variational autoencoder to align the distributions of the source domain and the target domain, which can be used as an alternative to the adversarial learning-based method of which the training is difficult and unstable. The third method is developed on the basis of a novel conception of learning the intra-domain style-invariant representation which aims to produce features invariant to the style diversity within the target domain thereby improving the generalization.

In Chapter 5, in the direction of model adaptation, we proposed two methods for multi-source model adaptation of semantic segmentation, of which one is for the general multi-source setting and the other one for the union-set multi-source setting. The union-set multi-source setting allows the label spaces of the source domains to be subsets of that of the target domain and is thus applicable to a wider range of scenarios. The methods for the two settings are both developed on the basis of a novel conception of model-invariant feature learning which harmonizes the model characteristics derived from different source domains to produce more generalizable features in the target domain.

The contributions of this thesis are the proposals of the methods in three different directions for mitigating the label dependence, which aim to improve the practicality of the deep learning-based image recognition technologies. The methods solve the problems remained in the previous studies, and the effectiveness of each method has been validated by conducting extensive experiments.

6.2 Future Directions

As the directions of future studies, we think that applications of unsupervised domain adaptation and model adaptation in fields such as medical imaging are worth paying more attention to. The domain shift problem is very common in medical images due to different scanners and imaging protocols. In this thesis, we evaluate our methods for unsupervised domain adaptation

and model adaptation with only street-scene images, and the effectiveness of the methods for medical imaging applications remains to be validated. Moreover, for the multi-source model adaptation methods presented in Chapter 5, because we used the same network structure for all the source-domain models in all the experiments, the effectiveness of the methods using different network structures has not been validated. Using source-domain models with different structures is more practical in real-world applications, and it may be necessary to adapt the proposed methods to the source-domain models with different structures.

Acknowledgments

First, I would like to sincerely thank my supervisor, Prof. Miki Haseyama. This thesis would not have been possible without the invaluable guidance and encouragement she has given me over the five years I spent at the Graduate School of Information Science and Technology, Hokkaido University.

I would like to thank Specially Appointed Prof. Kenji Araki, Specially Appointed Prof. Yuji Sakamoto, and Prof. Yoshinori Dobashi for providing insightful comments and suggestions about the research I performed at the Graduate School of Information Science and Technology, Hokkaido University.

I would also like to sincerely thank Prof. Takahiro Ogawa for the countless hours of assistance and fruitful discussion over the course of performing the work described in this thesis.

Furthermore, I would like to sincerely thank Specially Appointed Assistant Prof. Ren Togo for his tremendous support from the beginning of the research on this thesis and thank Specially Appointed Assistant Prof. Keisuke Maeda and Assistant Prof. Naoki Saito for their constant encouragement and advice about research and academic life.

Finally, I would like to sincerely thank everyone at the Laboratory of Media Dynamics, Graduate School of Information Science and Technology, Hokkaido University for their invaluable support and assistance.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[3] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[5] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[6] Y. Tokozume, Y. Ushiku, and T. Harada, "Between-class learning for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5486–5494.

[7] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3339–3348.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[9] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[10] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[11] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[13] A. Krause, P. Perona, and R. Gomes, "Discriminative clustering by regularized information maximization," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[14] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[15] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, vol. 41, no. 8, pp. 1979–1993, 2018.

[17] S. Park, J. Park, S.-J. Shin, and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[18] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[19] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, A. Solin, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *Neural Networks*, vol. 145, pp. 90–106, 2022.

[20] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[21] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2015, pp. 97–105.

[22] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[23] Z. He and L. Zhang, "Multi-adversarial faster-rcnn for unrestricted object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6668–6677.

[24] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6956–6965.

[25] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, "Adapting object detectors via selective cross-domain alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 687–696.

[26] Z. He and L. Zhang, "Domain adaptive object detection via asymmetric tri-way faster-rcnn," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 309–324.

[27] C. Chen, Z. Zheng, X. Ding, Y. Huang, and Q. Dou, "Harmonizing transferability and discriminability for adapting object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8869–8878.

[28] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 749–757.

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[30] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready, "A robust learning approach to domain adaptive object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 480–490.

[31] G. Zhao, G. Li, R. Xu, and L. Lin, "Collaborative training between region proposal localization and classification for domain adaptive object detection," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 86–102.

[32] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. Duan, and T. Yao, "Exploring object relation in mean teacher for cross-domain detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 457–11 466.

[33] J. Deng, W. Li, Y. Chen, and L. Duan, "Unbiased mean teacher for cross-domain object

detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4091–4101.

[34] Y. Zheng, D. Huang, S. Liu, and Y. Wang, "Cross-domain object detection through coarse-to-fine feature adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 766–13 775.

[35] M. Xu, H. Wang, B. Ni, Q. Tian, and W. Zhang, "Cross-domain detection via graph-induced prototype alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 355–12 364.

[36] C.-D. Xu, X.-R. Zhao, X. Jin, and X.-S. Wei, "Exploring categorical regularization for domain adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 724–11 733.

[37] Z. Zhao, Y. Guo, H. Shen, and J. Ye, "Adaptive object detection with dual multi-label prediction," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 54–69.

[38] 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 *Proceedings of the International Conference on Machine Learning*. PMLR, 2018, pp. 1989–1998.

[39] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7472–7481.

[40] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[41] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," *arXiv preprint arXiv:1603.03417*, 2016.

[42] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4085–4095.

[43] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 289–305.

[44] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.

[45] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5982–5991.

[46] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2517–2526.

[47] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2090–2099.

[48] J. Choi, T. Kim, and C. Kim, "Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6830–6840.

[49] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6936–6945.

[50] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, "Model adaptation: Unsupervised domain adaptation without source data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9641–9650.

[51] J. Liang, D. Hu, and J. Feng, "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*. PMLR, 2020, pp. 6028–6039.

[52] S. Yang, Y. Wang, J. Van De Weijer, L. Herranz, and S. Jui, "Unsupervised domain adaptation without source data by casting a bait," *arXiv preprint arXiv:2010.12427*, vol. 1, no. 2, p. 5, 2020.

[53] H. Xia, H. Zhao, and Z. Ding, "Adaptive adversarial network for source-free domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9010–9019.

[54] N. Ding, Y. Xu, Y. Tang, C. Xu, Y. Wang, and D. Tao, "Source-free domain adaptation via distribution estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7212–7222.

[55] F. Wang, Z. Han, Y. Gong, and Y. Yin, "Exploring domain-invariant parameters for source free domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7151–7160.

[56] J. Liang, D. Hu, J. Feng, and R. He, "Dine: Domain adaptation from single and multiple black-box predictors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8003–8013.

[57] S. Stan and M. Rostami, "Unsupervised model adaptation for continual semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2593–2601.

[58] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition, 2021, pp. 1215–1224.

[59] M. Ye, J. Zhang, J. Ouyang, and D. Yuan, "Source data-free unsupervised domain adaptation for semantic segmentation," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 2233–2242.

[60] J. Huang, D. Guan, A. Xiao, and S. Lu, "Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3635–3649, 2021.

[61] F. Fleuret *et al.*, "Uncertainty reduction for model adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9613–9623.

[62] N. Yamamichi, C. Hirano, M. Ichinose, Y. Takahashi, C. Minatsuki, R. Matsuda, C. Nakayama, T. Shimamoto, S. Kodashima, S. Ono *et al.*, "Atrophic gastritis and enlarged gastric folds diagnosed by double-contrast upper gastrointestinal barium x-ray radiography are useful to predict future gastric cancer development based on the 3-year prospective observation," *Gastric Cancer*, vol. 19, no. 3, pp. 1016–1022, 2016.

[63] R. Togo, N. Yamamichi, K. Mabe, Y. Takahashi, C. Takeuchi, M. Kato, N. Sakamoto, K. Ishihara, T. Ogawa, and M. Haseyama, "Detection of gastritis by a deep convolutional neural network from double-contrast upper gastrointestinal barium x-ray radiography," *Journal of Gastroenterology*, pp. 1–9, 2018.

[64] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[65] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in Statistics*. Springer, 1992, pp. 569–593.

[66] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[68] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, 2017, p. 3.

[69] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9627–9636.

[70] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[71] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[72] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[73] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2015, pp. 1180–1189.

[74] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[75] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, 2018.

[76] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[77] M. Wrenninge and J. Unger, "Synscapes: A photorealistic synthetic dataset for street scene parsing," *arXiv preprint arXiv:1810.08705*, 2018.

[78] C.-C. Hsu, Y.-H. Tsai, Y.-Y. Lin, and M.-H. Yang, "Every pixel matters: Center-aware feature alignment for domain adaptive object detector," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 733–748.

[79] D. Guan, J. Huang, A. Xiao, S. Lu, and Y. Cao, "Uncertainty-aware unsupervised domain adaptation in object detection," *IEEE Transactions on Multimedia*, 2021.

[80] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

[81] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. 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*, 2019, pp. 2507–2516.

[82] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 102–118.

[83] 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 *Proceedings of the European Conference on Computer Vision*, 2018, pp. 518–534.

[84] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured

output via discriminative patch representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1456–1465.

[85] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, "All about structure: Adapting structural information across domains for boosting semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1900–1909.

[86] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[87] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1791–1800.

[88] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 982–991.

[89] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 172–189.

[90] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[91] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[92] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[93] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[94] Q. Zhang, J. Zhang, W. Liu, and D. Tao, "Category anchor-guided unsupervised domain adaptation for semantic segmentation," in *Advances in Neural Information Processing Systems*, 2019, pp. 435–445.

[95] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 635–12 644.

[96] F. Lv, T. Liang, X. Chen, and G. Lin, "Cross-domain semantic segmentation via domain-invariant interactive relation transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4334–4343.

[97] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 975–12 984.

[98] Y. Yang, D. Lao, G. Sundaramoorthi, and S. Soatto, "Phase consistent ecological domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9011–9020.

[99] S. M. Ahmed, D. S. Raychaudhuri, S. Paul, S. Oymak, and A. K. Roy-Chowdhury, "Unsupervised multi-source domain adaptation without access to source data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 103–10 112.

Achievements of the Author

Papers

(A) Journal Papers

[**A-1**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Chronic gastritis classification using gastric X-ray images with a semi-supervised learning method based on tri-training," *Medical & Biological Engineering & Computing*, vol. 58, pp. 1239-1250, 2020. (2021 IF=3.079)

[**A-2**] Zongyao Li, Kazuhiro Kitajima, Kenji Hirata, Ren Togo, Junki Takenaka, Yasuo Miyoshi, Kohsuke Kudo, Takahiro Ogawa, Miki Haseyama, "Preliminary study of AI-assisted diagnosis using FDG-PET/CT for axillary lymph node metastasis in patients with breast cancer," *EJNMMI Research*, vol. 11, no. 1, pp. 1-10, 2021. (2021 IF=3.434)

[**A-3**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Learning intra-domain style-invariant representation for unsupervised domain adaptation of semantic segmentation," *Pattern Recognition*, vol. 132, 108911, 2022. (2021 IF=8.518)

[**A-4**] Zongyao Li, Keisuke Maeda, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Developing technologies for the practical application of deep learning-based distress segmentation in subway tunnel images," *Intelligence, Informatics and Infrastructure*, vol. 4, no. 1, pp. 1-8, 2023.

(B) International Conferences

[**B-1**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Semi-supervised learning based on tri-training for gastritis classification using gastric X-ray images," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, 2019.

[**B-2**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Classification of subcellular protein patterns in human cells with transfer learning," in *Proceedings of IEEE Global Conference on Life Sciences and Technologies (LifeTech)*, pp. 273-274, 2019.

[**B-3**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Unsupervised domain adaptation for semantic segmentation with symmetric adaptation consistency," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2263-2267, 2020.

[**B-4**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Variational autoencoder based unsupervised domain adaptation for semantic segmentation," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 2426-2430, 2020.

[**B-5**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Semantic-aware unpaired image-to-image translation for urban scene images," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2150-2154, 2021.

[**B-6**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Divergence-guided feature alignment for cross-domain object detection," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2240-2244, 2022.

[**B-7**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Improving model adaptation for semantic segmentation by learning model-invariant features with multiple source-domain models," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 421-425, 2022.

[**B-8**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, "Union-set multi-source model adaptation for semantic segmentation," in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 579-595, 2022.

(C) **Technical Report**

[**C-1**] 李 宗曜, 藤後 廉, 小川 貴弘, 平田 健司, 真鍋 治, 志賀 哲, 長谷山 美紀, “3D residual network に基づく FDG-PET/CT 画像を用いた悪性腫瘍候補の自動検出,” 映像情報メディア学会技術報告, vol. 43, no. 5, pp. 311-314, 2019.

[**C-2**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, “A note on retrieval of visually similar distress regions in subway tunnel images -Introduction of deep features extracted by semantic segmentation network-,” 映像情報メディア学会技術報告, vol. 44, no. 6, pp. 65-68, 2020.

[**C-3**] Zongyao Li, Ren Togo, Kenji Hirata, Kazuhiro Kitajima, Junki Takenaka, Yasuo Miyoshi, Kohsuke Kudo, Takahiro Ogawa, Miki Haseyama, “Detecting axillary lymph node metastasis of breast cancer with FDG-PET/CT images based on attention mechanism,” 映像情報メディア学会技術報告, vol. 45, no. 4, pp. 33-36, 2021.

[**C-4**] 李 宗曜, 藤後 廉, 小川 貴弘, 長谷山 美紀, “セマンティックセグメンテーションに対するマルチソースモデル適応に関する検討 -複数のソースモデルからの不変な特徴表現の学習による適応精度の向上-,” 映像情報メディア学会技術報告, vol. 46, no. 6, pp. 37-41, 2022.

[**C-5**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, “Union-set model adaptation for semantic segmentation using multiple source domains with subset label spaces,” 画像の認識・理解シンポジウム (MIRU), 2022.

[**C-6**] Zongyao Li, Ren Togo, Takahiro Ogawa, Miki Haseyama, “Source-data-free Domain-adaptive Semantic Segmentation with Inter-domain and Intra-domain Style Transfer,” 画像の認識・理解シンポジウム (MIRU), 2023.

(D) **Lecture**

[**D-1**] 李 宗曜, 藤後 廉, 小川 貴弘, 長谷山 美紀, “Tri-training に基づく胃 X 線画像を用いた胃炎の識別に関する検討,” 電気・情報関係学会北海道支部連合大会講演論文集, pp. 130-131, 2018.

(E) **Awards**

[E-1] 北海道大学大学院情報科学院 学院長賞 (2020 年 9 月)

[E-2] 画像の認識・理解シンポジウム 2022 学生奨励賞 (2022 年 7 月)