Train semantic segmentation neural networks with scarce annotated data

Linfeng Yuan

Submitted in fulfilment of the requirements for the
Degree of MSc by Research

School of Computing Science
College of Science and Engineering
University of Glasgow

April 2023

Abstract

Semantic segmentation is a challenging high-level vision task and high segmentation accuracy has been achieved with large quantities of well-annotated data. To reasonably reduce the demands for extensive pixel-level annotations, researchers conduct experiments in both semi-supervised learning (SSL) and unsupervised domain adaptation (UDA). They try to solve the problem of training semantic segmentation models with scarce annotated data from the perspective of different data distributions. The former addresses the issue of limited labeled data through knowledge transfer between data with independent and identical distributions, while the latter focuses on reducing the gaps and transfering knowledge between data with different distributions. Although both can alleviate the shortage of annotated data, they also face different problems. First of all, existing SSL only considers the feedforward neural network structure and ignores the crucial feedback mechanisms in the human visual system. This thesis investigates whether we can introduce feedback mechanisms into current feed-forward segmentation networks to improve segmentation accuracy or reduce the annotation requirements. We introduce two different feedback mechanisms and successfully depressing irrelevant background pixels by feeding the output mask back to the input end in many cases. In terms of UDA, current self-training-based UDA methods require a large-scale Graphics Processing Units (GPU) cluster for training which makes it impossible to conduct experiments with a single GPU. This project takes a multi-card state-of-the-art UDA framework as the baseline and modifies it as a single-card framework. Meanwhile, we introduce a re-sampling strategy which focuses the model on rare categories during training stage. Our proposed method gains similar segmentation accuracy to state-of-the-art frameworks with less GPU RAM during training. What's more, it significantly outperforms other current frameworks given less computational budgets. Through the above two aspects of research, this thesis provides a solution on how to train semantic segmentation networks with scarce annotated data.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to greatly thank my supervisors, Dr. Nicolas Pugeault and Dr. Jan Paul Siebert, for their supervision and support. Without their help, I cannot complete my master's degree during such a difficult period. I am also grateful to all the members of the CVAS group (Gerardo, Li, Ozan, Ali, Nikos,Vanja, Piotr, George, Daniella, Yu, etc). I feel motivated once I discuss and share with you. Thank you, UofG! And thank you, School of Computing Science! I will cherish all the memories of studying here.

Declaration

With the exception of chapters 1, 2 and 3, which contain introductory material, all work in this thesis was carried out by the author unless otherwise explicitly stated.

Chapter 1

Introduction

1.1 Motivation

With the development of Convolutional Neural Networks (CNN), many high level computer vision tasks such as image classification, object detection and semantic segmentation have attained high accuracy with fully supervised training approaches. Compared to image classification and object detection, semantic segmentation is the most challenging task whose goal is to assign a specific categorical label to each pixel in an image (i.e. classify all the pixels in the input image, as shown in the right picture of Figure 1.1). Given a raw image, semantic segmentation models aim to generate a precise and reasonable pixel-wise semantic map covering both foreground and background regions. Due to the dense scene information it gives, this task is an important component for a wide range of applications including robotic manipulation, autonomous driving, medical image analysis, and remote sensing image understanding. For the driving scenarios, the pixel-wise predictions of semantics can provide autonomous vehicles with information about driving areas (lane lines, roads) and obstacles (vehicles, pedestrians) to help them better complete scene perception and path planning. Benefiting from the large-scale datasets [13,16], many CNN-based semantic segmentation frameworks [9,31,39,52] gain significant improvements on accuracy. Given enough training data, state-of-the-art models are capable of producing high-quality predictions. Although high segmentation performance can be achieved when large quantities of annotated training data are available, labeling every pixel in images is time-consuming and expensive. In practice, the unavailability of pixel-wise annotations of high resolution images has been the largest and commonest bottleneck in fully supervised semantic segmentation. For example, labeling an image with 1024×2048 pixels in Cityscapes [13] will take more than 90 minutes which makes it impractical to prepare large-scale and fine-grained annotations for every semantic segmentation application. Therefore, extending the success of supervised semantic segmentation to domains where labeled data are scarce is an important problem.

One solution to this problem is to train segmentation models with limited data annotations. In

Figure 1.1: **Illustration of pictures from the synthetic domain, real scenes domain, and ground truth label for semantic segmentation.**

semi-supervised learning (SSL), a small portion of the labeled data is used in conjunction with large amounts of unlabeled data. Both labeled and unlabeled data are independent and identically distributed. Most recent SSL approaches are based on self-training techniques [10, 65]. These methods generate *pseudo labels* for unlabeled data in order to retrain the network with more annotated data. Another solution is unsupervised domain adaptation (UDA). It aims to adapt the model trained with a labeled source domain to an unlabeled target domain with acceptable accuracy loss. Different from the manually annotated target datasets [13, 16], the images from the source domain are usually data collected from video game engines, simulation platforms or rendering systems [38, 40]. These simulation tools significantly reduce the annotation period leading to very little annotation time, about 7 seconds for a frame on average. As a result, the challenges of UDA come from the huge domain gaps between the labeled source dataset and the unlabeled target dataset. As shown in the left and middle pictures of Figure 1.1, the left image comes from video game engines while the middle one is from the real scenes gathered by sensors. They look quite different although both of them display urban road scenes. Motivated by the development of semi-supervised techniques, self-training-based UDA methods have been given more attention and gain improvements for segmentation [2, 58]. These self-training-based UDA methods achieve more significant improvements in segmentation accuracy by generating pseudo labels for unlabeled target domain data.

There is evidence that the human visual system is not entirely feedforward but also makes use of feedback loops between regions [17]. The feedback mechanisms help human brains to acquire knowledge with only a few samples. However, deep neural networks usually rely on large amounts of data annotations. Only a few works focus on introducing feedback mechanisms to feedforward deep neural networks in recognition tasks [6, 43]. Most prevailing CNNs consist of only feedforward structures without feedback mechanisms to transmit high-level information to the low-level layers. Some works [34, 54] try to directly utilize feedback mechanisms to learn knowledge in unsupervised manners and fail. To this end, we aim to investigate the influence of feedback loops on fully supervised learning in order to bring their advantages to semi-supervised learning.

In terms of another solution to train models with scarce annotations, state-of-the-art UDA models have been widely studied and can reach high segmentation accuracy even if the real scene data are all unlabeled. They can achieve nearly 70% of the segmentation scores that fully supervised models gain. Although getting significant improvements, they require a huge number of computational budgets during training. The long training duration and high requirements of hardware significantly reduce the experimental efficiency and obstruct the development of UDA. In order to solve the problem, we modify a state-of-the-art UDA framework and reduce the training requirements of computational budgets while keeping segmentation accuracy by introducing a re-sampling strategy. The proposed method also outperforms state-of-the-art frameworks given same computational budgets during training.

1.2 Research Questions

This thesis aims to design a semantic segmentation framework that can gain high accuracy when annotations are scarce. In order to comprehensively achieve this goal, we investigate two research scenarios: whether or not the labeled and unlabeled data are in the same distribution. The former corresponds to the field of semi-supervised learning (SSL), while the latter corresponds to the research content of unsupervised domain adaptation (UDA). Through investigation and combination of these two fields, researchers can effectively promote the learning and optimization of semantic segmentation networks when data annotations are scarce. Specifically, we propose **RQ1** and **RQ2** for SSL, and **RQ3** and **RQ4** for UDA. Here are four research questions in this thesis.

- **RQ1:** Can recursive feedback mechanisms which feed mask prediction back to the original image improve segmentation accuracy?
- **RQ2:** Whether the feedback mechanisms raised in **RQ1** can work in semi-supervised learning approaches where training data are relatively limited?
- **RQ3:** Can self-training-based UDA frameworks be trained with limited computational resources?
- **RQ4:** Whether there is a significant drop in UDA accuracy when computational budgets are reduced? Can we solve this problem without introducing extra requirements of manpower and computational budgets?

1.3 Achievements

- We implement multiplicative feedback (MFB) and additional input channels feedback (ACFB) mechanisms with feed-forward CNNs. Although the simple feedback pipelines do not bring significant improvements to supervised learning, they can well suppress the background pixels in most cases.
- We introduce two different down-sampling methods of cropping windows into data augmentation pre-processing of ProDA [58]. Although the accuracy loss is relatively high, these methods significantly reduce the required number of GPUs, 50%-75% of computational resources are saved compared to ProDA. Meanwhile, we analyze the reasons for accuracy loss in order to get similar results to ProDA.
- In order to reduce the decreased accuracy caused by the smaller cropping windows, we introduce a *pseudo-label-guided* re-sampling strategy to focus the model on small and rare categories. The experimental results show that this strategy can significantly improve the segmentation accuracy on the hard categories which contributes a lot to the final segmentation scores.
- Finally, we come up with a UDA framework, ProDA-PLG, which requires only a single GPU. With only 25% of the computational budgets of ProDA, it gains similar segmentation accuracy to state-of-the-art UDA frameworks, which significantly improves experimental efficiency.

1.4 Overview

- **Chapter 1** introduces the problem and importance of this field. It also briefly points out my motivation, research questions and achievements.
- **Chapter 2** presents the background of related works on research topics for this thesis.
- **Chapter 3** provides details about used datasets and evaluation metric.
- **Chapter 4** introduces two different feedback mechanisms into the current segmentation network and illustrates the experimental results.
- **Chapter 5** gives basic information about a state-of-the-art UDA framework and gives two methods to reduce the requirements of computational resources. Meanwhile, we illustrate the qualitative and quantitative analysis for these two approaches and give the reasons for accuracy loss.

- **Chapter 6** raises a *pseudo-label-guided* method to mine the small and rare foreground instances in order to reduce the accuracy loss caused by the reduced computational budgets.
- **Chapter 7** summarizes the experiments of this dissertation and proposes future research plan.

Chapter 2

Literature Review

This chapter shows the literature review of relevant fields of my thesis. It would gradually introduce basic and state-of-the-art semantic segmentation algorithms, feedback mechanisms in neural networks, semi-supervised learning and Unsupervised Domain Adaptation for segmentation.

2.1 Semantic Segmentation

Most currently prevailing deep neural networks for semantic segmentation are based on fully convolutional network(FCN) [31] proposed in 2015. The high-dimensional feature maps generated by FCN don't contain fine-grained details because of the low resolution. It is caused by strided convolution and max pooling layers. Therefore, the accuracy of final predictions is not sufficient. In order to keep the resolution of feature maps before the pixel-wise classifier module, the encoder-decoder structures [3, 39] are widely studied in which the encoder is responsible for extracting features of images with down-sampling manners while the decoder restores the resolution and lost information. It also contributes to aggregating the most discriminative features also. Ronneberger et al. [39] propose a simple yet efficient encoder-decoder model, additionally using skip connections to pass information from low-level feature maps to high-level feature maps of the decoder. U-Net++ [62] introduces deep supervision and flexibly dense connections to U-Net [39]. Other remarkable works, DeepLab series [9, 29], replace the max pooling or strided convolutional layers with dilated convolutional layers in order to

keep the resolution of deep feature maps while enlarging the receptive field. The Atrous Spatial Pyramid Pooling structure allows the segmentation model to aggregate the features from different scales which drastically expands the receptive field of models though bringing extra computation budgets. Combining the characteristics of dilated convolution [29], improved ASPP [7, 8], encoder-decoder structure [39] and powerful backbone [12], DeepLabV3+ [9] has been regarded as the most powerful and robust CNN-based model for fully-supervised semantic segmentation task. Moreover, there are many methods gaining remarkable segmentation accuracy by keeping high resolution [46, 52], incorporating spatial context [60] or including object context [55,56]. Although those semantic segmentation models show powerful performance, the pixel-wise annotation process of large-scale datasets is time-consuming and complicated. Humans can learn from only a few samples due to the sophisticated connections between neurons in their brains [17]. In this thesis, feedback mechanisms are introduced to the fully-supervised deep neural network.

2.2 Feedback Mechanisms for Image Classification and Semantic Segmentation

Feedback connections are fundamental mechanisms in human brains and there have been several works to introduce feedback mechanisms to computer vision tasks using RNN, LSTM or top-down manners [6, 43, 44, 57]. Feedback network [57] uses convLSTM [43] to update hidden states with high-level features and provide feedback with input images. R2U-Net [1] is composed of a U-Net [39] where each convolutional layer is replaced by recurrent convolutional layer [28]. Feedback U-Net [44] is a U-Net [39] based segmentation model in which some of the standard convolutional layers are replaced by convLSTM [43]. Additionally, it introduces feedback from the segmentation predictions to the input end of the neural network. Different from the RNN-based or LSTM-based methods, U-Net with feedback attention mechanism [49] is an RNN-free framework utilizing standard convolutional layers to extract visual features. It uses a feedback mechanism from the last convolutional blocks to the first convolutional blocks, feeding the last convolutional layers back to the first ones as attention. Although those methods achieve some accuracy improvements in a fully supervised manner, no works focus on unsupervised learning based on feedback mechanisms. To this end, some researchers [34, 54] try to directly feed the predictions of CNNs back to the raw images in order to suppress irrelevant information. These methods train the segmentation neural networks in an unsupervised manner and get trivial results. Since feedback loops are proved to be useful in human visual learning, we assume that it is more reasonable to go from fully supervised to semi-supervised experiments. For this rea-

son, we investigate the impact of two different feedback mechanisms on semantic segmentation models in supervised learning to verify whether feedback mechanisms can suppress background information and explore the potential to utilize them in semi-supervised learning.

2.3 Semi-supervised Image Classification

Semi-supervised image classification has been widely studied before semi-supervised semantic segmentation. Many methods, e.g., temporal ensembling [27] and mean teacher [47], ensure consistency over different kinds of distortions or augmentations letting the models generate similar predictions or features with different views of the same image. By simultaneously updating two classifier networks initialized differently, dual student [24] imposes consistency between the outputs from differently augmented images. These methods can be regarded as performing consistency regularization. Moreover, there are many state-of-the-art methods combining consistency regularization and entropy minimization together. These methods introduce perturbations and generate artificial labels for unlabeled data, e.g., MixMatch [5] simultaneously considering different outputs on multiple kinds of augmented images, ReMixMatch [4] introducing distribution alignment and augmentation anchor to vanilla MixMatch [5] and FixMatch [45] utilizing pseudo labels on weakly augmented images to guide the predictions on strongly augmented pictures. Many of the improvements in semi-supervised image classification are then utilized in semi-supervised or unsupervised domain adaptation semantic segmentation, which will be introduced below. It is worth mentioning that the FixMatch [45] scheme has been utilized in our baseline UDA method [58].

2.4 Semi-supervised Semantic Segmentation

Benefiting from semi-supervised classification, consistency regularization and self-training techniques are also widely studied for semi-supervised semantic segmentation. The application of consistency regularization in semantic segmentation is to enforce the consistency of the predictions with different kinds of distortions at different levels such as input perturbations with different augmentations [18, 25], feature perturbations [37] or network perturbations by training two different classifiers. Self-training techniques for semi-supervised semantic segmentation [10, 65] mainly generate pixel-level pseudo labels on unlabeled images by models trained on labeled data after post-processing such as multi-sources fusion or filtering by threshold. The

process can be regarded as an iterative procedure in which there are 'generating pseudo labels' and 'retraining the model' in one training round. Some methods [23, 36] use the discriminator of GAN [19] learned for distinguishing the semantic predictions and labels to select predictions on unlabeled data as pseudo labels instead of simply filtering by threshold. PseudoSeg [65] also utilizes FixMatch [45] scheme using the pseudo predictions of weakly augmented images to guide the strongly augmented images. Nevertheless, CPS [10] adopts two same but independent networks, uses the predictions of each network to supervise another network and updates them simultaneously. Although these methods gain state-of-the-art segmentation accuracy, they only utilize feed-forward structures during training and inference. In this thesis, we explore the possibility of using feedback mechanisms in semantic segmentation.

2.5 Unsupervised Domain Adaptation for Semantic Segmentation

The predominant methods of UDA contribute to reducing the domain discrepancy by minimizing the divergence [32] or performing adversarial learning [19] at image levels [11, 21], feature levels [22, 42] or output space levels [48]. Recently, some methods originating from semi-supervised learning bring significant improvements for UDA. The entropy minimization [41,50] and self-training [2,58] have recently been regarded as simple yet powerful techniques for UDA applications. To decrease the over-confident wrong predictions caused by simple entropy minimization, CBST [63] utilizes an iterative pseudo labels generation scheme with a gradually increased portion of selected pseudo labels for each category. Due to the noise introduced by pseudo labels, CRST [64] additionally introduces regularization items at prediction and label levels while Seg-Uncertainty [61] explicitly generates the confidence maps of predictions to reduce the side-effect of unstable pseudo labels. CAG-UDA [59] estimates pseudo labels according to class centroids and performs feature alignment at the feature level. Nevertheless, these methods are optimized in an alternative scheme in which pseudo labels are fixed in a single training part. Recently, ProDA [58] and SAC [2] is proposed where the pseudo labels are updated on-the-fly. ProDA [58] incorporates prototypes to generate pseudo labels and performs consistency regularization in latent space while SAC [2] gets pseudo labels by multi-sources fusion of predictions from different augmentations of a single image. These state-of-the-art methods are sensitive to batch size and image resolution, so their segmentation scores are heavily dependent on the computational budgets. This leads to very bad results for training with a single GPU. Our aim is to explore a reasonable method to reduce the requirements of computing resources and use a method that does not introduce extra training time and computing power to

improve the final accuracy.

Chapter 3

Evaluation Method

This chapter describes the used datasets and the evaluation metric in this thesis.

3.1 Datasets

3.1.1 Customized COCO2017

For the training and evaluation of U-Net with feedback mechanisms, a customized subset of MS COCO [30] is utilized. The reason we use this dataset is that it contains many natural scenes with many foreground objects and corresponding detailed annotations. The original dataset involves 80 classes and a single image might contain multiple foreground instances belonging to different categories. Our U-Net with feedback mechanisms framework is designed as a class-agnostic semantic segmentation method. So we just consider 20 classes from the full dataset, re-sample 340 images for each class and ensure all the selected images only contain one foreground category as much as possible. The foreground categories include 'airplane', 'banana', 'bear', 'bicycle', 'bird', 'boat', 'bus', 'cat', 'cow', 'dog', 'donut', 'elephant', 'fire hydrant', 'giraffe', 'horse', 'motorcycle', 'sheep', 'stop sign', 'train' and 'zebra'. After selecting all the images, we resize the raw images and ground truth labels to 224×224 resolution. Then we set all the pixels belonging to the foreground category to 1 to generate binary masks for all the images. Among those 340 images for each class, 300 of them extracted from the training set of COCO act as

Figure 3.1: **Illustration of images and annotations in customized COCO2017.** The classes in this figure are respectively 'plane' and 'bicycle' in COCO2017.

training samples and 40 of them gathered from the validation set are regarded as validation sets. Figure 3.1 shows some examples of images and annotations from this dataset.

3.1.2 GTA5 and Cityscapes

Our UDA for semantic segmentation framework adapts the semantic segmentation model from synthetic source domain GTA5 [38] dataset to real scenes target domain Cityscapes [13] dataset. The reason for choosing these two datasets is that they share 19 categories, which are very common foreground and background objects in street scenes.

GTA5 dataset rendered from video games engine contains 24966 training images with the size of 1052×1914. We utilize 19 classes of GTA5 compatible with Cityscapes, including 'road', 'sideway', 'building', 'wall', 'fence', 'pole', 'light', 'sign', 'vegetation', 'terrace', 'sky', 'person', 'rider', 'car'. 'truck', 'bus', 'train', 'motorcycle' and 'bike'.

Cityscapes dataset gathered from real scenes of different cities contains 2975 training images and 500 validation images with the resolution of 1024×2048. It has pixel-wise annotations for 19 common categories same as GTA5 in different European cities. We regard the training images from Cityscapes as unlabeled data during training and evaluate our final model with

Figure 3.2: **Illustration of images and annotations in GTA5 and Cityscapes.** The first row shows the images and ground truth labels in the synthetic domain (GTA5). The second row illustrates the images and corresponding labels in the real scenes domain (Cityscapes).

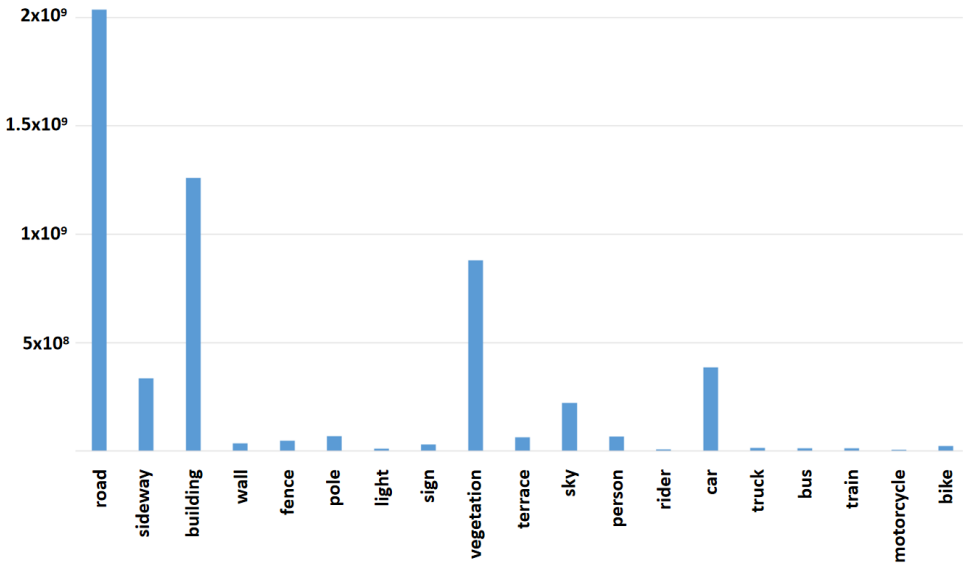

Figure 3.3: **Illustration of ground truth distributions for pixels in Cityscapes.** Obviously, the class distributions in Cityscapes are pretty imbalanced. In particular, the number of pixels belonging to 'motorcycle' is especially small compared to other categories(a quarter of pixels belonging to 'bicycle').

validation images from Cityscapes.

As shown in Figure 3.2, the two datasets share the same categorical labeling space while data distributions from them are not independent and identically distributed due to the illumination, camera pose, weather and huge domain gaps between video games and real scenes. What's more, as shown in Figure 3.3, the categories distributions of pixels in Cityscapes are extraordinarily imbalanced which makes it difficult to segment the pixels in UDA manner.

3.2 Evaluation Metrics

3.2.1 Intersection over Union

Intersection over Union (IOU) is an important evaluation metric for semantic segmentation. Similar to image classification, we can classify the predicted results of the segmentation model into 4 different categories. True Positive (TP) means the prediction and the label for the pixel are 'positive'. True Negative (TN) represents the prediction and the label for the pixel are 'negative'. False Positive (FP) is the wrong prediction for the 'negative' ground truth. And False Negative (FN) shows the label for the pixel is 'positive' while the prediction is the wrong class. IOU can be calculated by the equation below:

$$\text{IOU} = \frac{TP}{TP + FP + FN} \tag{3.1}$$

In the evaluation stage, the prediction and ground truth of each pixel in each testing sample are used to calculate the intersection and union. These results will be accumulated and used for the calculation of the final IOU of each category.

3.2.2 Mean IOU and Frequency Weighted IOU

After getting IOU for each category, the overall segmentation accuracy of the model on the test set can be evaluated by two metrics. Here we assume that there are K classes in the dataset. And

the IOU for each category can be denoted as IOU_k.

One metric for the overall accuracy of the model is mean IOU (mIOU), which is the mean value of all IOU_k belonging to all categories. This metric treats each class fairly, regardless of their proportion in the testing samples. It can be calculated as follow:

$$\text{mIOU} = \sum_{k=1}^{K} \frac{\text{IOU}_k}{K} \tag{3.2}$$

Another metric, named frequency weighted IOU (fwIOU), can also be used to evaluate the overall accuracy of the model. Different from mIOU, fwIOU uses the frequency of pixels belonging to each category to appear in the testing samples to weigh the IOU_k. Therefore, this general metric will be biased towards those categories with a high proportion in the testing samples, while ignoring some rare categories. It can be calculated as follow:

$$\text{fwIOU} = \sum_{k=1}^{K} f_k \text{IOU}_k \tag{3.3}$$

where f_k indicates the frequency weight of category k, and $\sum_{k=1}^{K} f_k = 1$.

For most semantic segmentation frameworks, mIOU is the main evaluation metric. But it treats each category fairly, so when the model performs poorly on a very small number of categories, the final mIOU will be greatly affected. For some tasks that care about the general segmentation accuracy and do not pursue some rare categories, fwIOU will also be considered. In this thesis, mIOU will be calculated to evaluate the accuracy of our segmentation models.

Chapter 4

U-Net with Feedback Mechanisms

In order to efficiently utilize identically distributed annotated and unlabeled data to train segmentation models, we try to introduce feedback mechanisms to the fully-supervised and semi-supervised deep neural networks. In this chapter, we raise two research questions: 1) can recursive feedback mechanisms which feed mask prediction back to the original image improve segmentation accuracy? 2) whether the feedback mechanisms can work in semi-supervised learning approaches where training data are relatively limited. This chapter introduces two feedback mechanisms, multiplicative feedback (MFB) and additional input channels feedback (ACFB) mechanisms, into a fully-supervised semantic segmentation framework. Although both of them do not bring a significant improvement to the segmentation accuracy, they can succeed in suppressing the background while keeping the foreground pixels in most cases. We analyze the success and failure cases for future studies.

4.1 U-Net with Feedback Mechanisms

In order to answer the first research question, we propose a fully-supervised segmentation framework combining U-Net [39], variational autoencoder [26] and two different schemes of feedback mechanisms together. The reason we choose U-Net is that its structure is very simple and it is not prone to overfitting. Feedback loops play a crucial role in the human visual system. If a reasonable feedback loop framework can be established, the segmentation model will also have the ability to learn more critical knowledge from a small number of samples. We hope to estab-

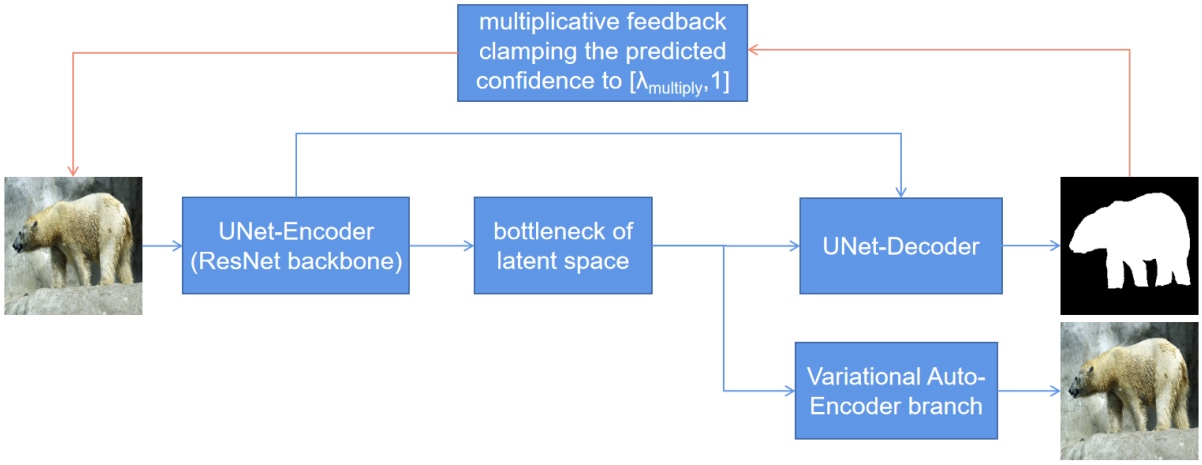

Figure 4.1: **Illustration of U-Net with the multiplicative feedback mechanism (MFB).** The predicted hard confidence maps of the segmentation model would be clamped to $[0.9, 1]$ before the multiplication of original images and clamped predictions so that the irrelevant pixels are gradually suppressed and tend to 0 after $T = 10$ steps.

lish feedback mechanisms from the output end to the input end to tell the model the foreground and background pixels it predicted in the previous step so that it can obtain better prediction results and learning capabilities. The proposed feedback mechanisms, namely MFB and ACFB, iteratively generate mask predictions and feed them back to the original images. Specifically, the multiplicative feedback mechanism (MFB) directly suppresses the background pixels in the original images while the additional input channels feedback mechanism (ACFB) regards the generated masks as additional input channels. Back to the research question, if the proposed feedback mechanisms can perform well in a fully-supervised manner. We can adapt the frameworks to semi-supervised ones in order to answer the second research question. Unfortunately, our proposed MFB and ACFB cannot improve the IOU scores in a fully-supervised manner. So we analyze the failure reasons and the cases of generated masks in the feedback loops.

4.1.1 Multiplicative Feedback Mechanism

Figure 4.1 simply illustrates the U-Net with variational autoencoder and multiplicative feedback mechanism. In this setting, we would run the feedforward (blue lines in Figure 4.1) and feedback (orange lines in Figure 4.1) loops for $T = 10$ times. For each loop, we multiply the original images and the output maps where all the negative pixel-wise predictions are replaced by $\lambda_{multiply} = 0.9$ to gradually suppress the background. We simply utilize direct multiplication because of the assumption that the predicted confidence in a fully-supervised learning scheme

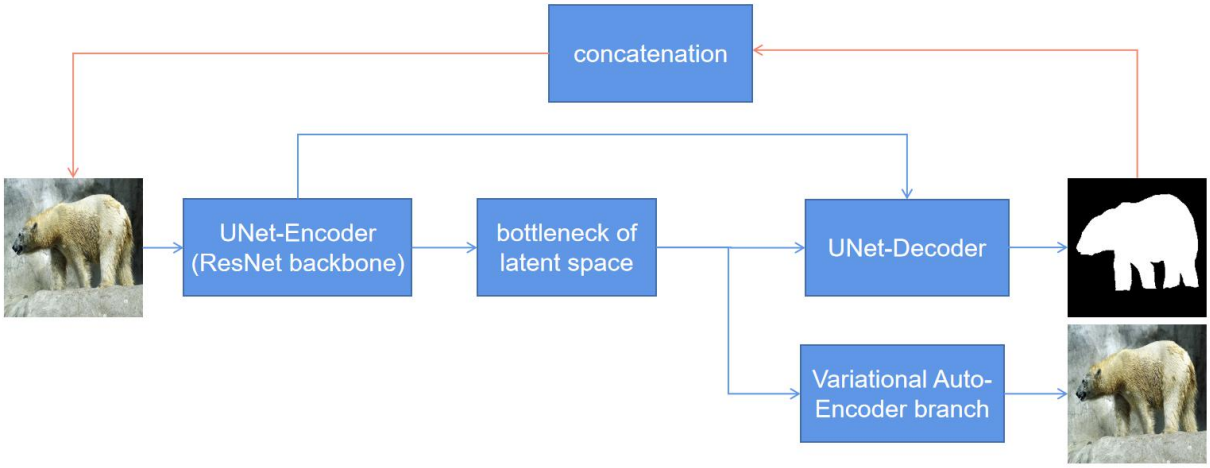

Figure 4.2: **Illustration of U-Net with feedback from output end to input end as additional input channels.** The new predictions from the model in each step would be concatenated to the original images before the next step proceeds, letting the model learn how to feed the output back to the original images.

is relatively accurate so that we do not need to adjust it like pseudo labels in self-training. In this combined model, the variational autoencoder performs the image reconstruction task which is supervised with pixel-wise mean-squared-error loss and KL loss. The back-propagation of reconstruction and KL losses can improve the discriminative ability of the image encoder. The details of the proposed MFB are as follow:

Algorithm 1 Proposed Multiplicative Feedback Mechanism (MFB)

Input: The input batch of images, I; The height and width of input images, (H, W); The constant multiplicative factor, $\lambda_{multiply} = 0.9$; The number of feedback iterations $T = 10$.

Output: The generated mask predictions M and reconstructed images I' for input images I.

$M, I' \leftarrow model(I)$;
for $t \leftarrow 2$ to T **do**
  for $i \leftarrow 1$ to H **do**
    for $j \leftarrow 1$ to W **do**
      $M_{ij} \leftarrow M_{ij} * \lambda_{multiply}$;
    end for
  end for
  $I \leftarrow I * M$;
  $M, I' \leftarrow model(I)$;
end for
return M and I';

4.1.2 Additional Input Channels Feedback Mechanism

Figure 4.2 displays the U-Net with variational autoencoder and feedback from output to input end. In this scheme, we would also run the feedforward (blue lines in Figure 4.2) and feedback (orange lines in Figure 4.2) loops for $T = 10$ times. Instead of multiplying original images and clamped hard confidence maps, we introduce the feedback from the output end as additional information entered into the model by concatenation. The reason for directly using the hard segmentation predictions is that we hypothesize that the predictions are accurate enough and what we need to do is to feed them back to the input end in order to let the model learn how to combine the original images and previous predictions. The details of the proposed ACFB are as follow:

Algorithm 2 Proposed Additional Input Channels Feedback Mechanism (ACFB)

Input: The input batch of images, I; The single channel 0 tensor which has the same spatial dimension as input images, Z; The number of feedback iterations $T = 10$.

Output: The generated mask predictions M and reconstructed images I' for input images I.

$I_c \leftarrow concatenate(I, Z)$;
$M, I' \leftarrow model(I_c)$;
for $t \leftarrow 2$ to T **do**
    $I_c \leftarrow concatenate(I, M)$;
    $M, I' \leftarrow model(I_c)$;
end for
return M and I';

4.2 Experiment

4.2.1 Implementations

As mentioned in Chapter 3, the dataset we select is a customized subset of MSCOCO [30]. We select 20 classes as the foreground objects. And 300 images for each category are sampled for training while 40 images are selected for testing. This project is conducted on a Tesla V100 GPU whose memory is 16 GB with PyTorch 1.6 implementation. We utilize U-Net [39] for the semantic segmentation model with the backbone ResNet-18 [20]. All the images are resized to 224×224. For both multiplicative and additional input channels feedback mechanisms, we select Adam optimizer with initialized learning rate as 0.0005 which is decayed by 0.8 every 20

Model	mIOU
U-Net	82.11
U-Net with MFB	79.64
U-Net with ACFB	**82.14**

Table 4.1: Comparison results of customized MSCOCO segmentation in terms of mIOU (U-Net, U-Net with MFB and U-Net with ACFB). The best score of each column is highlighted.

Model	mIOU
DeepLabV2	83.19
DeepLabV2 with MFB	80.35
DeepLabV2 with ACFB	**83.21**

Table 4.2: Comparison results of customized MSCOCO segmentation in terms of mIOU (DeepLabV2, DeepLabV2 with MFB and DeepLabV2 with ACFB). The best score of each column is highlighted.

epochs. The weight decay is set to 0.0002.

4.2.2 Experiment Results

We conduct experiments on U-Net with multiplicative feedback (MFB) and additional input channels feedback mechanisms (ACFB) compared to standard U-Net. As shown in Table 4.1, the proposed MFB and ACFB cannot bring significant improvements in accuracy on segmentation. What's worse, the U-Net with multiplicative feedback (MFB) shows an IOU drop of 2.47 mIOU. Compared to the standard U-Net getting 82.11 mIOU, the U-Net with multiplicative (MFB) and additional input channels feedback(ACFB) mechanisms respectively get 79.64 and 82.14 mIOU. The IOU drop of U-Net with MFB may result from directly suppressing the background pixels in the input images. Some high-level visual features(e.g., global context, relations between the foreground and background pixels) are lost because only foreground pixels remain. And in some cases, these visual clues play an important role in the segmentation process. So for future studies of the MFB, we can consider to suppress the background pixels in the feature space according to the predicted masks. Since each point in the features maps have aggregated enough local features, suppressing the background and highlighting the foreground pixels in the feature space can eliminate irrelevant information while keeping the global context of the images. In terms of the U-Net with ACFB, it shows similar IOU scores compared to the standard U-Net. And the predicted masks in each iteration are nearly the same as each other, indicating that directly concatenating the predicted masks and input images is not a proper feedback mechanism. In addition to U-Net, we also integrate the two proposed feedback mechanisms into the DeepLabV2 [7]. As shown in Table 4.2, the impact of these two feedback mechanisms on DeepLabV2 is overall consistent to their impact on U-Net.

In addition to the quantitative analysis on mIOU, we also show a lot of qualitative analysis on U-Net with MFB. Figure 4.3, 4.4, 4.5, 4.6 respectively illustrate the inference process of U-Net

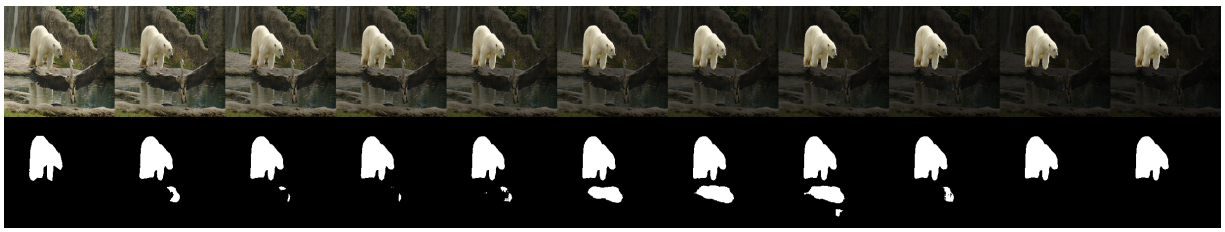

Figure 4.3: **Illustration of good cases with multiplicative feedback.** The first column contains respectively original image and ground truth labels. The following columns represent images and predictions in different time steps.

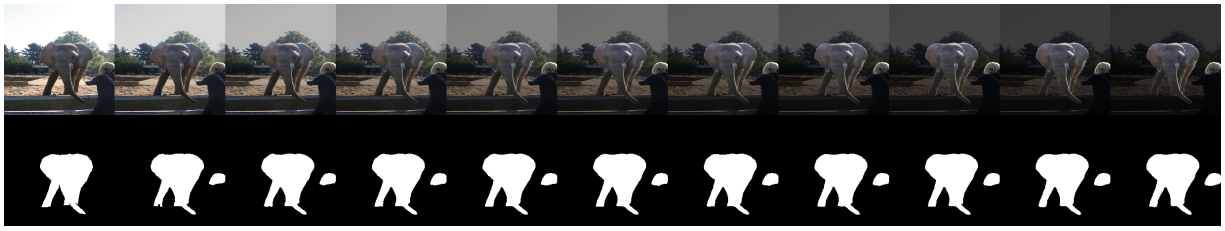

Figure 4.4: **Illustration of normal cases with multiplicative feedback.** The first column contains respectively original image and ground truth labels. The following columns represent images and predictions in different time steps.

with MFB on different testing images step-by-step. We can observe as the time step iterates, what predictions the model will produce and how it will affect the input image of the next time step.

Figure 4.3 shows a good case that MFB helps the model eliminate wrong predictions. The U-Net model cannot distinguish the foreground pixels well at the first iteration. There are some false positive (FP) pixels on the right of the polar bear. But with the iterative variation of the original images, the background pixels are suppressed and only pixels belonging to the polar bear are kept finally. The predictions are good in this kind of variation. However, in most cases,

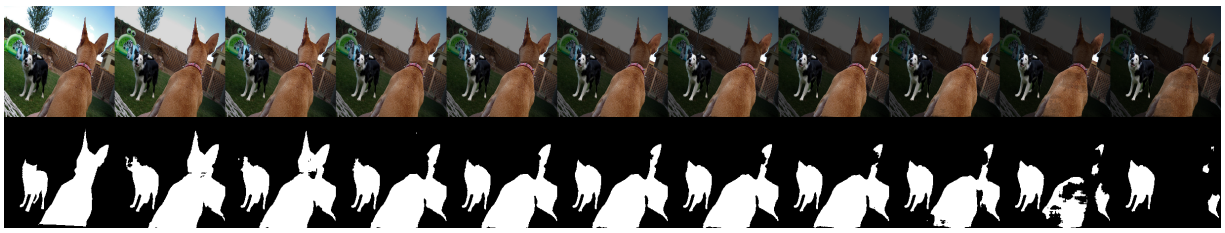

Figure 4.5: **Illustration of bad cases with multiplicative feedback.** The first column contains respectively original image and ground truth labels. The following columns represent images and predictions in different time steps.

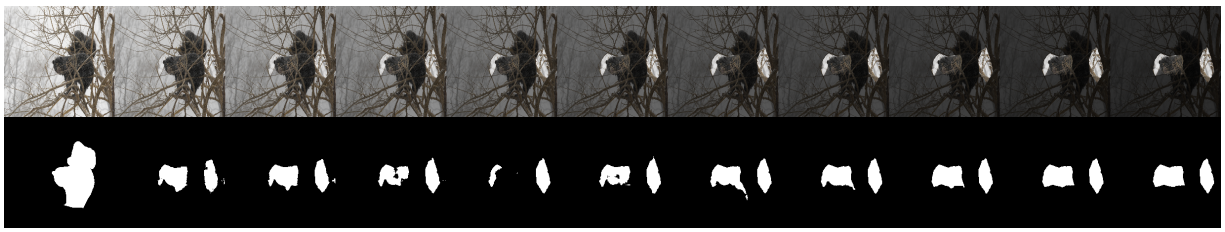

Figure 4.6: **Illustration of very bad cases with multiplicative feedback.** The first column contains respectively original image and ground truth labels. The following columns represent images and predictions in different time steps.

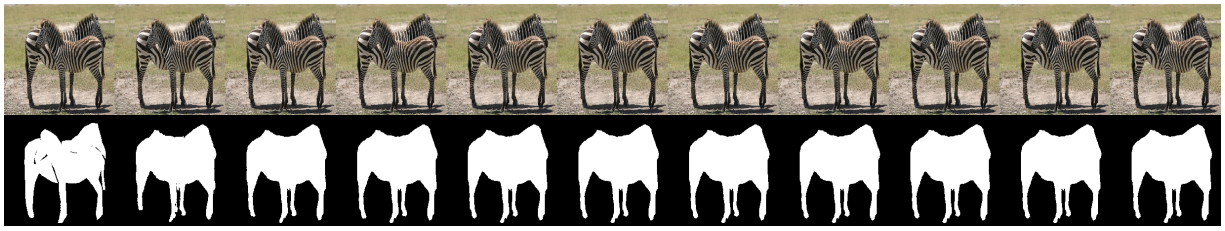

Figure 4.7: **Illustration of normal cases with additional input channels feedback.** The first column contains respectively original image and ground truth labels. The following columns represent images and predictions in different time steps.

the iterative variations are similar to that shown in figure 4.4 in which the false positive pixels are kept so that the pixels belonging to FP samples would be retained after the last iteration. Moreover, Figure 4.5 and Figure 4.6 show what happens in a few hard cases. Because of some false negative predictions (FN) predictions, the information about the brown dog in Figure 4.5 is gradually missed as the iterative process proceeds. At the end of the iterations, the pixels containing the dog disappear a lot so that the model cannot distinguish the dog at all. In Figure 4.6, the situation is much worse that the model cannot distinguish the foreground at the first iteration so that the wrong pixels are retained and taken into consideration permanently. These bad cases lead to the decreased IOU scores of U-Net with MFB.

We also show an example of the iterative predictions of U-Net with additional input channels feedback (ACFB) mechanism, as shown in Figure 4.7. The binary prediction at each iteration is almost constant. In other words, the number of iterations of time steps in ACFB does not affect the prediction of the model. And the final mIOU of U-Net with ACFB is nearly the same as that of the standard U-Net (82.14 vs 82.11). It means that simply concatenating the images with the mask predictions of the previous iteration cannot bring significant improvements.

4.3 Conclusion

In order to answer the first research question, we introduce two different feedback mechanisms, namely MFB and ACFB, into a fully supervised segmentation framework, U-Net. According to the experiment results, U-Net with MFB can successfully suppress the predicted background pixels with the iterations proceeding. However, in many cases, the false positive pixels remain while only in a few images they are filtered in the iterations. Meanwhile, the lost global context in some cases may lead to wrong mask predictions, which makes the final mIOU lower compared to the standard U-Net. In terms of U-Net with ACFB, the mask predictions are the same as those of U-Net. It proves that concatenating the segmentation maps from the previous iteration and input images is not a proper feedback mechanism. The feedback mechanisms from the output end to the input end do not work well in fully supervised segmentation. How to add a suitable denoising strategy to the feedback mechanism is a topic worthy of study in the future. In the remaining part of this thesis, we will introduce in detail how to use scarce annotated data to train segmentation models through knowledge transfer between data with different distributions.

Chapter 5

ProDA with Limited Computational Budgets

Unlike semi-supervised learning (SSL), which pays attention on knowledge transfer between annotated and unlabeled data with the same distribution, unsupervised domain adaptation (UDA) focuses on knowledge transfer between labeled and unlabeled data with different distributions. It addresses the challenge of limited annotations when training semantic segmentation models by mitigating discrepancies among various image domains. Therefore, the second half of this thesis focuses on the study of UDA methods to reduce the heavy reliance of segmentation models on data annotations. In this research field, self-training-based methods have achieved significant improvements in segmentation accuracy by generating pseudo labels for unlabeled data and denoising them. However, the segmentation accuracy of self-training-based UDA frameworks highly depends on the computational budgets. This raises the bar for experimentation and limits efficiency. So in this chapter, a research question is raised: Can self-training-based UDA frameworks be trained with limited computational resources? This chapter introduces basic knowledge of a state-of-the-art self-training-based UDA framework, ProDA [58]. To answer the research question, we introduce two strategies into the original data augmentation pipeline in ProDA to reduce the huge requirements of computing resources during training. The proposed ProDA-HR and ProDA-LR succeed in reducing the computational budgets but exhibit some problems which will be solved in Chapter 6.

5.1 Preliminaries of ProDA

5.1.1 Self-training Based UDA Segmentation

In section 5.1, ProDA [58] is introduced since it is a state-of-the-art UDA framework and we utilize it as a basic model of our work. In order to unify the mathematics symbols of this thesis, we uniformly use x_s to denote labeled data or data from source domain, and x_t to represent unlabeled data or data from target domain. Given labeled dataset $\mathscr{X}_s = \{x_s\}_{j=1}^{n_s}$ with corresponding ground truth labels $\mathscr{Y}_s = \{y_s\}_{j=1}^{n_s}$, we aim to train a semantic segmentation model learned from $\mathscr{X}_s$ and perform well on the unlabeled dataset $\mathscr{X}_t = \{x_t\}_{j=1}^{n_t}$ without accessing target ground truth labels $\mathscr{Y}_t$, in which $\mathscr{Y}_s$ and $\mathscr{Y}_t$ share the same K categories. Typically, the semantic segmentation model $h = g \circ f$ can be regarded as the combination of a backbone f and a classification module g.

Generally, the source network we get cannot generalize to unlabeled target data because of the domain discrepancy. To transfer the learned knowledge from labeled data to unlabeled data, conventional self-training-based methods CBST [63] and CRST [64] train the models with a standard cross-entropy loss under the supervision of pseudo labels $\hat{y}_t$:

$$\ell_{ce}^{t} = -\sum_{i=1}^{H\times W}\sum_{k=1}^{K} \hat{y}_t^{(i,k)} \log\left(p_t^{(i,k)}\right) \tag{5.1}$$

where $p_t = h(x_t)$ and $p_t^{(i,k)}$ denotes the activated soft-max probability of pixel x_t^i belonging to the kth category. Generally, we use the class with the highest confidence prediction for each pixel as pseudo labels:

$$\hat{y}_t^{(i,k)} = \begin{cases} 1, & \text{if } k = \arg\max_{k'} p_t^{(i,k')} \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

Here this transform can be denoted by $\hat{y}_t = \xi(p_t)$. Intuitively, directly leveraging the pseudo labels cannot gain good sgementation results because the pseudo labels are noisy. So researchers tend to select the pixels whose predicted confidence is higher than a threshold to retrain the models. Following this setting, the networks in target domain can learn from well-classified pixels, and then the updated pseudo labels with new predictions and thresholds can be used for the next stage.

5.1.2 Pseudo Labels Denoising Scheme

In our baseline, ProDA [58], the pseudo labels are updated online because updating pseudo labels after one training round are too late as the model has already over-fitted the noisy labels, while simultaneously updating the network parameters and pseudo labels is easy to get trivial solutions. So ProDA [58] freezes the initial pseudo labels generated by source model and gradually weights them by category-wise probabilities ω_t which are calculated before each iteration in conjunction with newly updated knowledge. So the weighted pseudo labels on the fly can be calculated as:

$$\hat{y}_t^{(i,k)} = \xi\left(\omega_t^{(i,k)} p_{t,0}^{(i,k)}\right) \tag{5.3}$$

where $\omega_t^{(i,k)}$ represents the weight for modifying frozen pseudo labels and $p_{t,0}^{i,k}$ is the fixed pseudo labels generated by source model.

Let $f(x_t)^{(i)}$ denote the feature of x_t at index i, $\tilde{f}(x_t)^{(i)}$ represent the feature of x_t at index i generated by an exponentially weighted moving average (EMWA) model referred to momentum encoder of feature extractor f, $\eta^{(k)}$ are the prototypes which represent categorical centroids in feature space.

$$\omega_t^{(i,k)} = \frac{\exp\left(-\left\|\tilde{f}(x_t)^{(i)} - \eta^{(k)}\right\|/\tau\right)}{\sum_{k'}\exp\left(-\left\|\tilde{f}(x_t)^{(i)} - \eta^{(k')}\right\|/\tau\right)} \tag{5.4}$$

where τ is the softmax temperature usually set to $\tau = 1$. The assumption is that the momentum feature extractor $\tilde{f}$ can generate more stable and independent predictions rather than the learnable model f. By calculating the distances between prototypes and features generated by a momentum feature extractor, we can use the distances to weight the fixed pseudo labels on-the-fly instead of directly retraining the network with noisy pseudo labels. The denoising process of pseudo labels is shown in Fig 5.1.

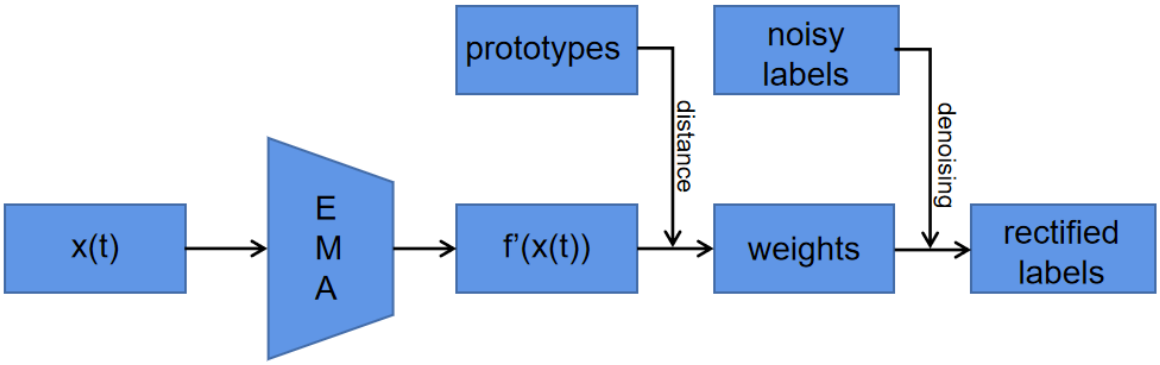

Figure 5.1: **Visualization of pseudo labels denoising process.**

5.1.3 Prototypes Calculation and Updating

Here comes the strategy for initializing and updating prototypes. According to the proposed label updating scheme [58], we require to compute the prototypes online. At the beginning of training, the prototypes $\eta^{(k)}$ are initialized as the average of features filtered by the pseudo labels $\hat{y}_t$ for all the images from target domain:

$$\eta^{(k)} = \frac{\sum_{x_t \in \mathscr{X}_t} \sum_i f(x_t)^{(i)} * \mathbb{1}\left(\hat{y}_t^{(i,k)} = 1\right)}{\sum_{x_t \in \mathscr{X}_t} \sum_i \mathbb{1}\left(\hat{y}_t^{(i,k)} == 1\right)} \tag{5.5}$$

where $\mathbb{1}$ is indicator function. In practice, the calculation is time-consuming during training which makes it impossible to update the prototypes on the whole target dataset. To solve the problem, the prototypes are evaluated as the moving average of class centroids in mini-batches. And in each iteration, the prototypes are updated by:

$$\eta^{(k)} \leftarrow \lambda \eta^{(k)} + (1-\lambda)\eta'^{(k)} \tag{5.6}$$

where $\eta'^{(k)}$ denotes the mean features of specific category k calculated with the current batch from momentum encoder. The momentum coefficient is set to 0.9999. And the shape of prototypes is $[n_{classes}, n_{features}]$ where $n_{features}$ is set to 256. So the prototype for each category is a 256-d vector which corresponds to the depth of output feature maps from feature extractor f and momentum feature extractor $\tilde{f}$.

5.1.4 Loss Function

After getting predictions and weighted pseudo labels, ProDA [58] uses symmetric cross-entropy loss [53] instead of standard cross-entropy loss to further enhance the noise tolerance to stabilize the early training phase:

$$\ell_{sce}^{t} = \alpha \ell_{ce}(p_t, \hat{y}_t) + \beta \ell_{ce}(\hat{y}_t, p_t) \tag{5.7}$$

where α and β are balancing coefficients and set to 0.1 and 1 respectively [53].

Besides the cross-entropy loss on source images and symmetric cross-entropy loss on target images, there are 2 items involved in the final objective function. Because of the domain gap, the target distribution generated by feature extractor is more likely to be dispersed so that the prototypes cannot succeed in rectifying the labels of pixels whose features lie at far end of cluster centroids even if the general target features can be well classified by source model. ProDA [58] aims to learn the underlying structure of target domain and obtain a more compact latent space that is friendly to the pseudo labels refinement. It utilizes generated features under weak augmentation to guide the learning for the strong augmented view of x_t. Concretely, let $\mathscr{T}(x_t)$ and $\mathscr{T}'(x_t)$ denote weakly and strongly augmented view for x_t respectively. We can compute the weak prototypical assignments $z_{\mathscr{T}}$ and strong prototypical assignments $z_{\mathscr{T}'}$ by:

$$z_{\mathscr{T}}^{(i,k)} = \frac{\exp\left(-\left\|\tilde{f}(\mathscr{T}(x_t))^{(i)} - \eta^{(k)}\right\|/\tau\right)}{\sum_{k'}\exp\left(-\left\|\tilde{f}(\mathscr{T}(x_t))^{(i)} - \eta^{(k')}\right\|/\tau\right)} \tag{5.8}$$

$$z_{\mathscr{T}'}^{(i,k)} = \frac{\exp\left(-\left\|f(\mathscr{T}'(x_t))^{(i)} - \eta^{(k)}\right\|/\tau\right)}{\sum_{k'}\exp\left(-\left\|f(\mathscr{T}'(x_t))^{(i)} - \eta^{(k')}\right\|/\tau\right)} \tag{5.9}$$

Since $z_{\mathscr{T}}$ is more stable and the input x_t suffers from fewer perturbations, we use it to guide learnable feature extractor f to generate consistent assignments for strongly augmented images:

$$\ell_{kl}^{t} = \mathrm{KL}(z_{\mathscr{T}} \| z_{\mathscr{T}'}) \tag{5.10}$$

This equation can help the model make consistent prototypical labeling leading to more compact latent space in target domain.

Since Equation 5.10 may result in compact feature space, some minor categories might become empty. A regularization term is introduced letting the predictions of the models be smoothly distributed:

$$\ell_{reg}^{t} = -\sum_{i=1}^{H\times W}\sum_{j=1}^{K}\log p_{t}^{(i,k)} \tag{5.11}$$

Finally, We can retrain the model with the final objective function:

$$\ell_{\text{total}} = \ell_{ce}^{s} + \ell_{sce}^{t} + \gamma_1\ell_{kl}^{t} + \gamma_2\ell_{reg}^{t} \tag{5.12}$$

where the γ_1 and γ_2 are set to 10, 0.1 respectively.

5.2 ProDA Trained with Limited Computational Budgets

As mentioned in Chapter 1, self-training-based UDA models for semantic segmentation need a huge number of computational resources and a large amount of time to train. For example, the duration of each training stage of ProDA [58] is more than 40 hours with 4 Tesla V100 whose memories are no less than 16GB in total while SAC [2] also needs 4 TitanX to train the models whose backbone is ResNet-101 [20]. Our goal is to train the self-training-based segmentation network with fewer computing resources such as training the network with only a single GPU whose memory is 16GB without a significant drop in IOU scores. To adapt to the limited computational budgets, we attempt to use two modified pre-processing methods to make the training stage executed with only one card.

In Section 5.1, we concretely describe the training process and objective function of ProDA [58]. And in this section, we will introduce two modified pre-processing methods to decrease the required number and memories of GPU cards. The data pre-processing method of ProDA [58] is composed of random scaling, cropping and horizontal flipping. We propose two variants

Method	Cropping Size	Resolution	Receptive Field
ProDA	$[H_{cropping}, W_{cropping}]$	R_{ProDA}	F_{ProDA}
ProDA-HR	$[H_{cropping}/2, W_{cropping}/2]$	R_{ProDA}	$F_{ProDA}/4$
ProDA-LR	$[H_{cropping}/2, W_{cropping}/2]$	$R_{ProDA}/4$	F_{ProDA}

Table 5.1: **Comparison of the different pre-processing methods.**

of the original setting to save the memory usage of the GPU. The first method, ProDA-HR (high resolution) is to fix the scale of the original images and directly decrease the cropped windows' size by a factor of 0.5. Intuitively, that would make the memory requirement of the GPU decrease to a quarter of the original one so that we can train the model with only one card. The fine-grained details in raw images are kept while the smaller cropped patches would not cover large spatial areas as the original ones. Therefore, we can automatically think of another method. The second method, ProDA-LR (low resolution), is to down-sample the original images by a factor of 0.5 before generating the smaller cropped training samples with the same size as those from ProDA-HR. Different from ProDA-HR, it can keep the general spatial information, but lose fine-grained details because of the down-sampling process of original images. Those two methods can generate training samples with a size of $[H_{cropping}/2, W_{cropping}/2]$ so that we can optimize the model with one single GPU instead of 4 cards. And the cropped windows sampled by these two methods represent two different aspects compared to the original ProDA, receptive fields and fine-grained details. Table 5.1 illustrates the difference between cropped windows under different pre-processing strategies.

5.3 Experiment

5.3.1 Implementations

GTA5 [38] and Cityscapes [13] are separately selected as the annotated source and unlabeled target datasets. The purpose of this experiment is to adapt the knowledge learned from GTA5 to Cityscapes without accessing the ground truth. We use DeepLabV2 [7] for the semantic segmentation model with the backbone ResNet-101 [20]. Before self-training proceeds, we utilize single-level AdaptSegNet [48] to perform adversarial learning at the output space level as the warm-up stage. During self-training-based optimization, we choose SGD optimizer with initialized learning rate as 0.0001 which is decayed by 0.9 every epoch. The total epochs are 84 while

Model	GPUs	road	sideway	building	wall	fence	pole	light	sign	vege	terrace	sky	person	rider	car	truck	bus	train	motor	bike	mIOU	gain
Source	-	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6	+0.0
AdaptSegNet	1	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	+4.8
CyCADA	1	86.7	35.6	80.1	19.8	17.5	38.0	39.9	41.5	82.7	27.9	73.6	64.9	19.0	65.0	12.0	28.6	4.5	31.1	42.0	42.7	+6.1
CLAN	1	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	+6.6
ADVENT	1	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5	+8.9
FADA	4	91.0	50.6	86.0	**43.4**	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	**38.0**	85.2	31.6	46.1	6.5	25.4	37.1	50.1	+13.5
CBST	1	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9	+9.3
MRKLD	1	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1	+10.5
CAG_UDA	1	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	+13.6
Seg-Uncertainty	1	90.4	31.2	85.1	36.9	25.6	37.5	**48.8**	**48.5**	85.3	34.8	81.1	64.4	36.8	86.3	34.9	52.2	1.7	29.0	44.6	50.3	+13.7
IAST	2	**93.8**	**57.8**	85.1	39.5	26.7	26.2	43.1	34.7	84.9	32.9	**88.0**	62.6	29.0	87.3	39.2	49.6	23.2	34.7	39.6	51.5	+14.9
SAC	4	90.4	53.9	**86.6**	42.4	27.3	45.1	48.5	42.7	**87.4**	40.1	86.1	67.5	29.7	**88.5**	**49.1**	**54.6**	9.8	26.6	45.3	**53.8**	**+17.2**
ProDA	4	91.6	51.8	83.1	41.8	**35.7**	40.1	44.1	43.4	87.1	**43.3**	79.6	66.5	31.6	86.8	40.8	53.2	0.0	**45.6**	52.8	53.6	+17.0
ProDA-HR	1	90.5	57.3	80.6	43.0	32.6	41.3	43.4	47.4	86.4	36.4	81.4	66.7	27.4	86.7	35.5	48.6	0.0	8.1	50.1	50.7	+14.1
ProDA-LR	1	92.1	54.3	81.8	32.5	25.7	**45.2**	34.2	40.8	84.9	30.3	76.7	**68.2**	22.1	87.6	31.9	39.3	0.0	24.5	**53.3**	48.7	+12.1

Table 5.2: **Comparison results of GTA5→Cityscapes adaptation in terms of mIoU.** These quantitative results of other UDA frameworks are taken from papers of ProDA [58] and SAC [2]. The number of GPUs for different methods is collected via GitHub. If the same method presents different scores in these two papers, we show it with the higher score. The best score for each column is highlighted.

the weight decay is set to 0.0002. Before training, the soft pseudo labels and initialized prototypes are pre-generated. In terms of the strong augmentation for consistency regularization, we select RandAugment [14] and CutOut [15]. For the weak data augmentation, the original based scaling size is 2200 with random factor from 0.5 to 1.5, and the original cropping size is 512×896. In order to save computational resources, ProDA-HR and ProDA-LR vary the cropping size directly to 256×448. Different from ProDA-HR, ProDA-LR sets the based resolution of raw images as 1100 in order to gain a bigger receptive field though losing fine-grained details.

5.3.2 Experiment Results

We compare the official ProDA [58] ProDA-LR and ProDA-HR with multiple famous and state-of-the-art methods. They can be divided into two types: 1) GAN-based methods which utilize adversarial learning to align the distribution, including AdaptSegNet [48], CycADA [21], CLAN [33], ADVENT [50], FADA [51]; 2) self-training based methods, including CBST [63], MRKLD [64], CAG_UDA [59], Seg-Uncertainty [61], IAST [35], SAC [2].

As shown in Table 5.2, the official ProDA method reaches state-of-the-art accuracy, 53.6 mIOU (without the knowledge distillation which can gain 4.0 mIOU additionally), outperforming all the methods on UDA from GTA5 to Cityscapes task except SAC [2]. In terms of the categorical IOU, the official ProDA, ProDA-HR and ProDA-LR achieve the best IOU in 6 categories among 19 classes in Cityscapes. And especially they gain the highest score on some small categories

such as 'fence', 'pole', 'person', 'motor', and 'bike'. Moreover, the high scores also occur on important and dynamic categories for autonomous driving such as 'person', 'car', 'truck', 'bus', 'motor', and 'bike'. According to the table, we can find that even if the computational resources are reduced, our ProDA-HR and ProDA-LR can gain outstanding IOU scores compared to other state-of-the-art methods given the same GPU.

However, the models trained with smaller cropped windows cannot recognize a few specific categories well. For instance, the ProDA-HR gets similarly fine-grained input images with smaller receptive fields corresponding to the original images compared to official ProDA settings. It cannot perfectly distinguish the 'terrace', 'rider', 'truck', 'bus' and 'motor' compared to the original setting. What's worse, the 'motor' category is ignored completely by the ProDA-HR. In another word, the 'motor' category vanishes in the latent space. It results from the fact that the 'motor' category is rare in the target domain. And the ProDA-HR samples a much smaller cropped window than the original ProDA which makes it much harder to let the model 'see' the 'motor' category. That's why the IOU score of the 'motor' class is low while testing, leading to lower overall segmentation accuracy and higher risk when guiding an autonomous vehicle.

The issues about the discriminant ability of the 'motor' category still exist in ProDA-LR but are much better than in ProDA-HR. Although ProDA-LR cannot get as high IOU score on the 'motor' as the original ProDA, it can gain similar one compared to other state-of-the-art models. What's more, it can gain high IOU scores on more than half categories (e.g. 'road', 'sideway', 'building', 'pole', 'vegetation', 'person', 'car' and 'bike') which are even better than original ProDA. This phenomenon proves that the training results of half of the classes in Cityscapes do not rely on high resolutions of raw images. Unfortunately, for some of the categories (e.g. 'wall', 'fence', 'traffic light', 'sign', 'truck' and 'bus'), the lost fine-grained details during training would lead to significant accuracy drop which lowers the final IOU scores.

These problems above are what we should solve in Chapter 6. In the rest part of this chapter, we will do qualitative research for ProDA-HR which will be used in Chapter 6.

As shown in Figure 5.2 and Figure 5.3, we can find that even though the pre-generated fixed pseudo labels would contain noises, the occurrence of prototypes can correctly denoise the pseudo labels generated by the source model. And after long-term training, the trainable model exceeds the slowly updated momentum moving average model. In conjunction with the IOU of the trained model, we can get the conclusion that ProDA can be successfully trained with limited computational budgets, getting high IOU scores except few categories compared to that trained with 4 Tesla V100 cards.

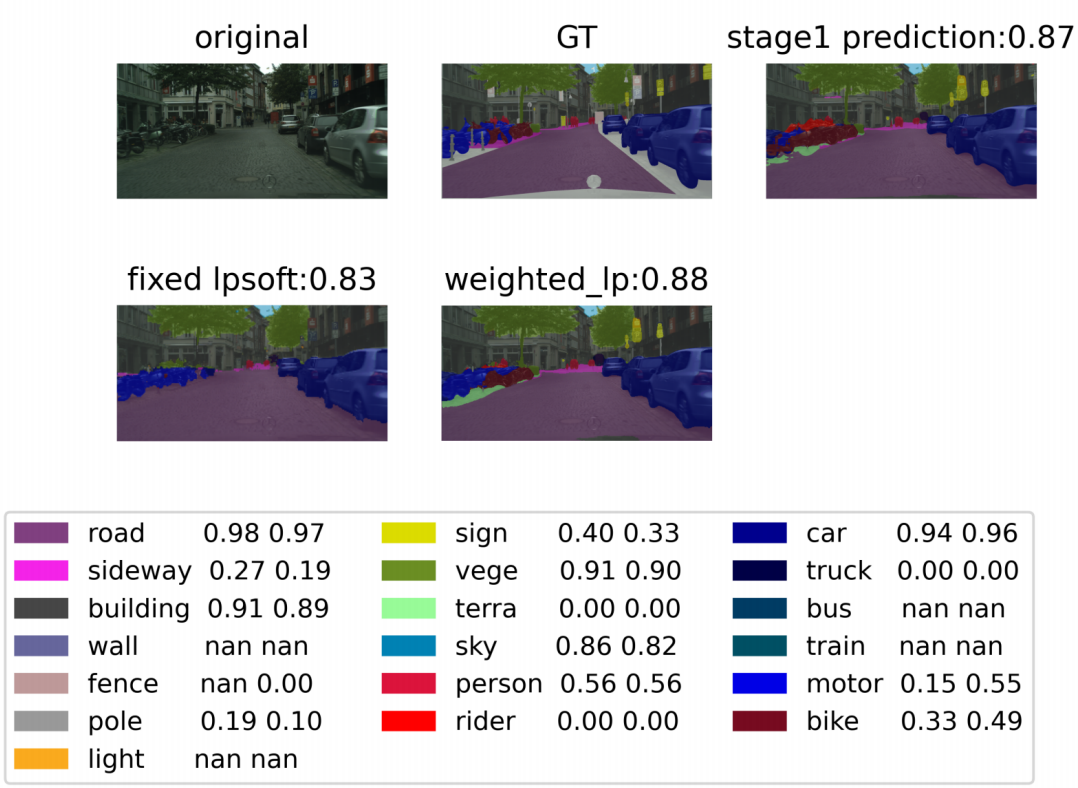

Figure 5.2: **Illustration of predictions, fixed pseudo labels, weighted pseudo labels.** From top left to bottom right, each image represents the original image, ground truth labels, predictions of the model, pre-generated fixed pseudo labels, and weighted pseudo labels after denoising. In the legends, the columns respectively represent the color of each category, the name of each category, the IOU of each category from the moving average model, and the IOU of each category from the trainable model.

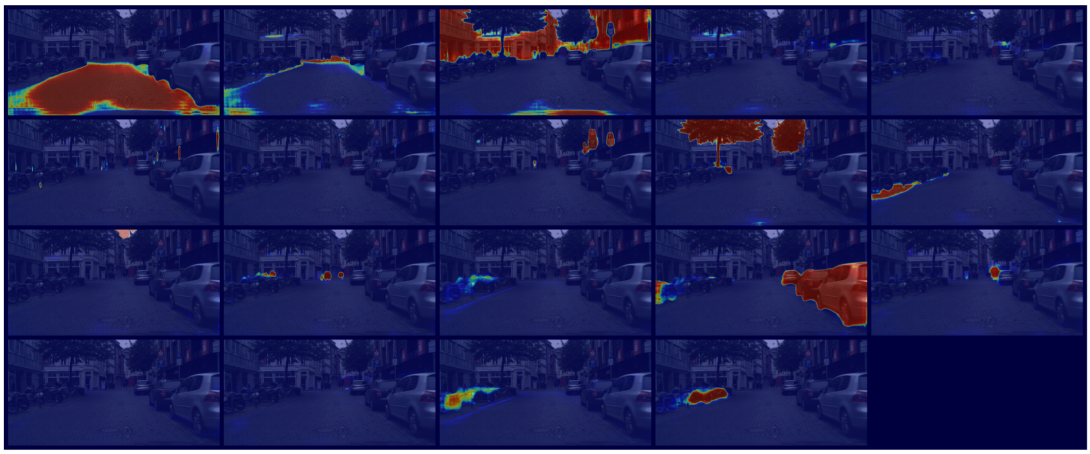

Figure 5.3: **Illustration of the weights for denoising the fixed noisy pseudo labels.** From the top left to the bottom right, each heat map represents the weight of a specific category for denoising the pre-generated fixed pseudo labels.

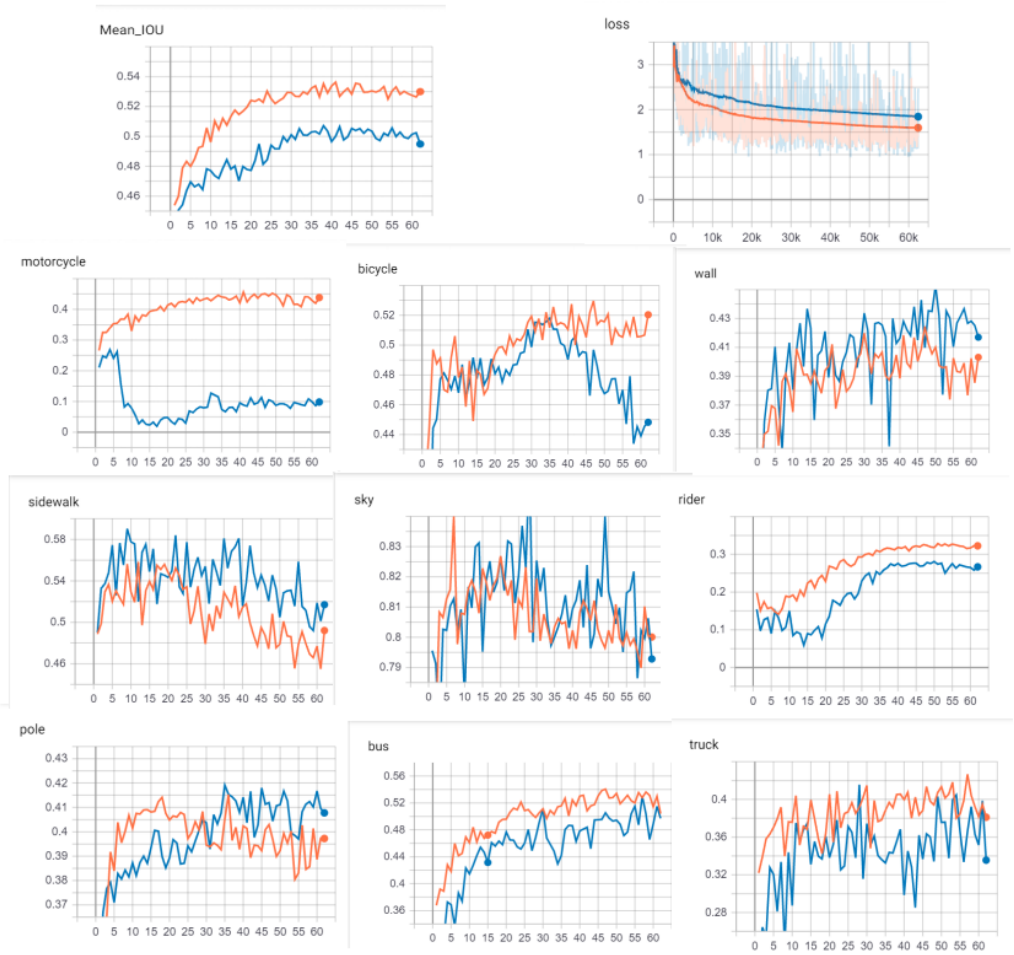

Figure 5.4: **Illustration of training curves of original ProDA and ProDA-HR.** The orange curves represent the original ProDA, and the blue curves represent ProDA-HR. The y-axis of the second picture is the overall loss of ProDA while those of other pictures are the absolute IOU values.

Finally, we will illustrate the training curves of the original ProDA and ProDA-HR. As shown in Figure 5.4, the mIOU of ProDA-HR is lower than the original ProDA because of the much smaller cropping size, but the curves also show that ProDA-HR can converge well except in the 'motor' category. It results from the bad recognition ability of motorcycles caused by missing pixels belonging to the 'motor' during training. With training proceeding, ProDA-HR gains higher IOU scores on 'wall', 'sideway' and 'pole' compared to the original ProDA. Meanwhile, it can keep a similar ability to recognize many categories.

5.4 Conclusion

In this chapter, we propose two different strategies to decrease the requirements of computational budgets of state-of-the-art ProDA [58]. These two methods are proven to be efficient and ProDA-HR can gain higher accuracy than that of state-of-the-art frameworks when given the same GPU. The problem of missing the 'motor' class in the proposed ProDA-HR results from the fact that 'motor' is scarce and small in the imbalanced target dataset. Although PorDA-HR confronts the problem of losing 'motor' category, it answers the research question raised in the introduction. By introducing a proper sampling strategy, we can save three-quarters of the required computational budgets of self-training-based UDA frameworks with an acceptable IOU drop. And the long-tail recognition problem is solved in the next chapter.

Chapter 6

ProDA with Pseudo-Label-Guided Re-sampling Strategy

In the previous chapter, we introduce ProDA-HR and ProDA-LR to decrease the requirements of computational budgets during training. The experimental results show that the proposed ProDA-HR saves 75% computational budgets during training and outweighs state-of-the-art UDA segmentation models given the same GPU. Since the target dataset is imbalanced and some objects are scarce and small in size, a few categories vanish in the latent space when the training process proceeds. So we raise a research question here: can we solve the problem of vanishing categories when computational budgets are reduced without introducing extra requirements of manpower and computational budgets? In order to solve the long-tail recognition problem, this chapter introduces a pseudo-label-guided (PLG) re-sampling strategy into ProDA-HR. Without extra manpower and computational resources, our ProDA-PLG trained with limited GPUs can get a similar segmentation accuracy to ProDA trained with 4 GPUs. With 25%-50% computational resources, the ProDA-PLG solves the problem of low accuracy on rare and small categories by a wiser hard-sample mining algorithm, leading to significant improvements in saving GPU RAM and increasing IOU scores over state-of-the-art methods given the same GPU.

Figure 6.1: **Illustration of evaluation results of ProDA-HR with random cropping strategy and ProDA-HR trained with our pseudo-label-guided re-sampling strategy.** The orange boxes highlight the confusing area containing overlapped 'motor', 'bike' and 'rider' classes. The ProDA-HR cannot distinguish the difference between 'motor' and 'bike' while the ProDA-PLG can generate more accurate predictions compared to the former one.

6.1 ProDA with Pseudo-Label-Guided Re-sampling Strategy

As described in the previous chapter, both of these two down-sampling strategies have disadvantages. For the ProDA-HR, we can keep the fine-grained details in raw images during training. But the down-sampled cropped windows are too small so it is difficult to cover some small and rare foreground instances such as 'motor'. With the iterations proceeding, some small and rare categories disappear in the latent space, leading to extremely low IOU scores while evaluation (as the bottom left picture shown in Fig 6.1). As for the ProDA-LR, the training samples cropped on the down-sampled raw images can cover larger receptive fields which makes it possible to mine the hard samples and objects compared to the ProDA-HR. However, the lost fine-grained information caused by the down-sampling strategy would lead to a significant accuracy drop on many categories which rely on the fine-grained details in high-resolution raw images.

Actually, it is hard to make a trade-off between hard sample mining and image resolution by

simply altering the resolution of raw images and cropped windows. Since image resolution and high-frequency information are crucial to semantic segmentation models, this chapter will focus on how to mine difficult sample information that disappears due to the use of smaller cropping window in ProDA-HR. Therefore, we propose a pseudo-label-guided re-sampling strategy namely PLG in order to force the data loader of the CNNs framework to focus on the hard and rare patches of pixels. Because lacking the annotations in target domains, we do not train an object detector for real scenarios as sampling priors. The proposed method of re-sampling strategy is to utilize the warm-up model in ProDA [58] which is responsible to generate fixed pseudo-labels during training. The generated fixed pseudo-labels can find the approximate positions of every category in images although the details are not precise enough. We will use them as prior information to guide the re-sampling strategy to mine the hard and rare samples in the target domain during training.

In the following sections, the pseudo-label-guided (PLG) re-sampling strategy is introduced in both text and algorithm.

Algorithm 3 Proposed PLG Re-sampling Strategy

Input: The selected priority queue for rare categories in target domain, Q_p; The number of selected rare categories, N_k; The set of unlabelled target images (full size) for the current batch, X_t; The set of soft pseudo labels (full size) for the current batch, P_t;

Output: The set of pairs including cropping windows containing rare categories and corresponding pseudo labels for training, $B_r = (X_r, P_r)$;

$X_r \leftarrow []$
$P_r \leftarrow []$
for each image and pseudo label $(x_t, p_t) \in (X_t, P_t)$ **do**
    Find the rarest pixel $p_{rarest} \in p_t$ according to Q_p and regard it as the center point;
    Crop the image patch around the p_{rarest} in (x_t, p_t) with random offsets to generate a cropping window x_r and its corresponding pseudo label p_r;
    $X_r \leftarrow concatenate(X_r, x_r)$;
    $P_r \leftarrow concatenate(P_r, p_r)$;
end for
$B_r \leftarrow (X_r, P_r)$
return B_r;

Before training, we regard N_k categories to be the hard classes with sequential priority in a priority queue Q_p. In this setting, N_k is a hyperparameter representing the length of priority queue Q_p. The selected classes and the order in Q_p can be decided in two ways. A) One way is based on manual selection. For example, we can manually select N_k categories belonging to dynamic foreground objects which are important in autonomous driving scenes and set the order of these classes in Q_p according to the importance of different classes for self-driving vehicles. B) Another solution is to choose N_k categories and the order in Q_p by prior information such as the fixed pseudo labels generated by the source warm-up model in ProDA [58]. Since the target

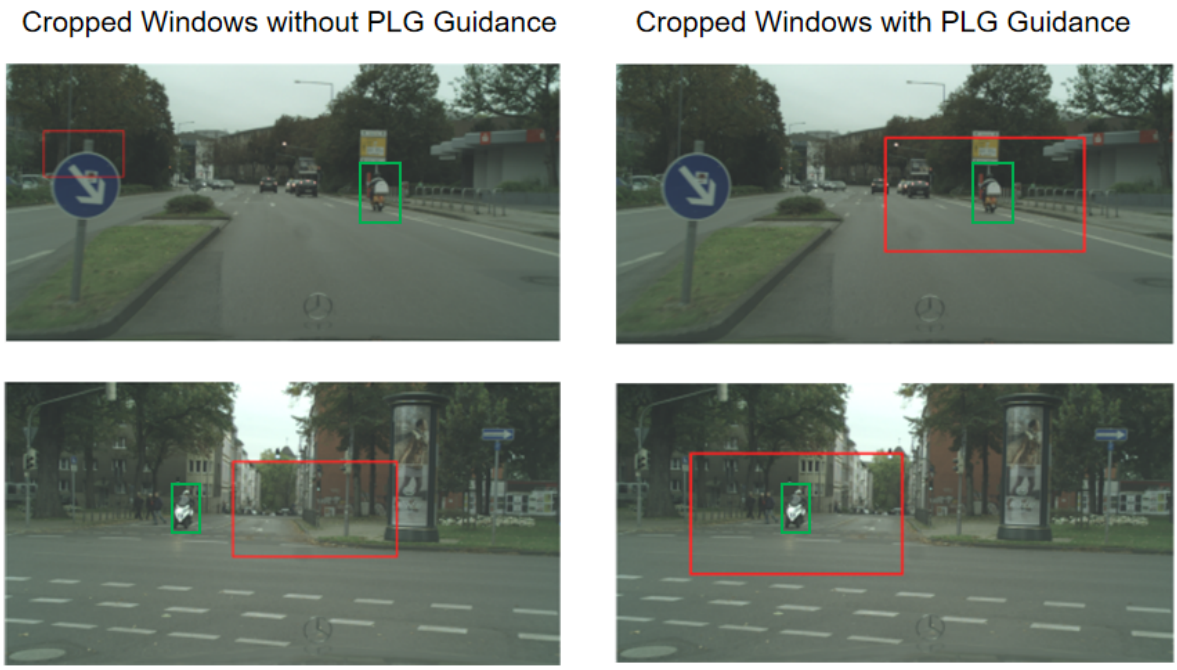

Figure 6.2: **Illustration of training windows to be cropped without/with our re-sampling strategy.** The red boxes represent the cropped windows for training. The green boxes highlight the position of defined rare classes. The comparison of the cropped windows with/without our proposed PLG method shows that the PLG method can solve the long-tail recognition problem for the imbalanced dataset by focusing more on specific categories during training.

dataset is imbalanced in 19 categories. We can run the warm-up model on the unlabeled target dataset to generate pseudo-labels and count the number of pixels belonging to different classes in all generated target semantic maps. Assuming that we take N_k categories as rare categories, the categories' priority in Q_p can be determined according to the number of predicted pixels belonging to specific rare categories. The fewer pixels that are predicted to belong to a rare class, the higher the priority of that class in the queue Q_p. Because a smaller number of pixels in output space means that the class is less numerous in the target domain or the warm-up model cannot recognize it well. In this way, we can leverage the prior information of the unlabeled target domain to tackle the long-tail recognition problem without accessing the labels. And the rest of the classes not belonging to the Q_p share the equally lowest priority because they are categories not important in the target domain or easy to learn by UDA methods.

After deciding the Q_p and N_k, the proposed PLG method directly affects the data loader during training. The data loader of ProDA-PLG tends to sample 'hard' areas with priority according to the Q_p and generated pseudo labels. And it can perform hard-sample mining to tackle the long-tail recognition problems by cropping windows covering rarer categories compared to the uniformly random cropping strategy. Even though sometimes the 'hard' cropped windows sam-

pled by this PLG re-sampling strategy do not actually contain the small and rare categories, it would not do harm to the final training results compared to random cropping sampling. What's more, the model and data are calculated on GPU while the proposed PLG method only affects the training data loader which utilizes CPU only, so the proposed method does not introduce extra running time also and can significantly mine the hard samples. Algorithm 3 provides a brief overview of how PLG works during training.

6.2 Experiment

6.2.1 Implementations

The task and used datasets of ProDA-PLG (ProDA-HR with pseudo-label-guided strategy) are the same as those of ProDA-HR and ProDA-LR. GTA5 [38] and Cityscapes [13] are separately selected as the annotated source and the unlabeled target datasets. The purpose of this experiment is to adapt the knowledge learned from GTA5 to Cityscapes without accessing the ground truth. In terms of the re-sampling method, we manually set the number of mining categories N_k to 2 and only select 'motorcycle' and 'bike' to be the rare categories with first and second highest priority because the number of pixels belonging to 'motor' in pseudo labels are much more limited rather than other categories, which means that the pseudo labels are incorrect on 'motor' class or the target dataset contains few pixels belonging to this category. In either case, it means that the category is difficult to recognize and deserves special attention. Before the data loader samples a training sample in target domains, the method would scan the pseudo labels to check whether there 'might' be hard samples in this raw image. If so, the data loader would select a proper patch covering the hard samples according to the priority queue. By this method, we can keep the high resolution of raw images and at the same time mine the hard samples as much as possible. Figure 6.2 demonstrates the effect of our re-sampling strategy. We can observe that the data loader seldom focuses on the hard samples containing rare pixels with a random cropping strategy. And with the proposed PLG re-sampling strategy, the model with a smaller cropping size can be trained with proper samples which contain the small and rare categories according to the priority queue.

Model	GPUs	road	sideway	building	wall	fence	pole	light	sign	vege	terrace	sky	person	rider	car	truck	bus	train	motor	bike	mIOU	gain
Source	-	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6	+0.0
ProDA	4	**91.6**	51.8	**83.1**	41.8	35.7	40.1	**44.1**	43.4	**87.1**	**43.3**	79.6	66.5	31.6	86.8	**40.8**	**53.2**	0.0	**45.6**	**52.8**	**53.6**	**+17.0**
ProDA-HR	1	90.5	**57.3**	80.6	**43.0**	32.6	41.3	43.4	**47.4**	86.4	36.4	**81.4**	66.7	27.4	86.7	35.5	48.6	0.0	8.1	50.1	50.7	+14.1
ProDA-PLG	1	89.6	55.0	79.5	37.9	**38.3**	**42.1**	**44.1**	42.8	87.2	36.4	80.2	**67.0**	**33.0**	**86.9**	33.3	46.5	0.0	37.0	49	51.9	+15.3

Table 6.1: Comparison results of GTA5→Cityscapes adaptation in terms of mIoU. The best score for each column is highlighted.

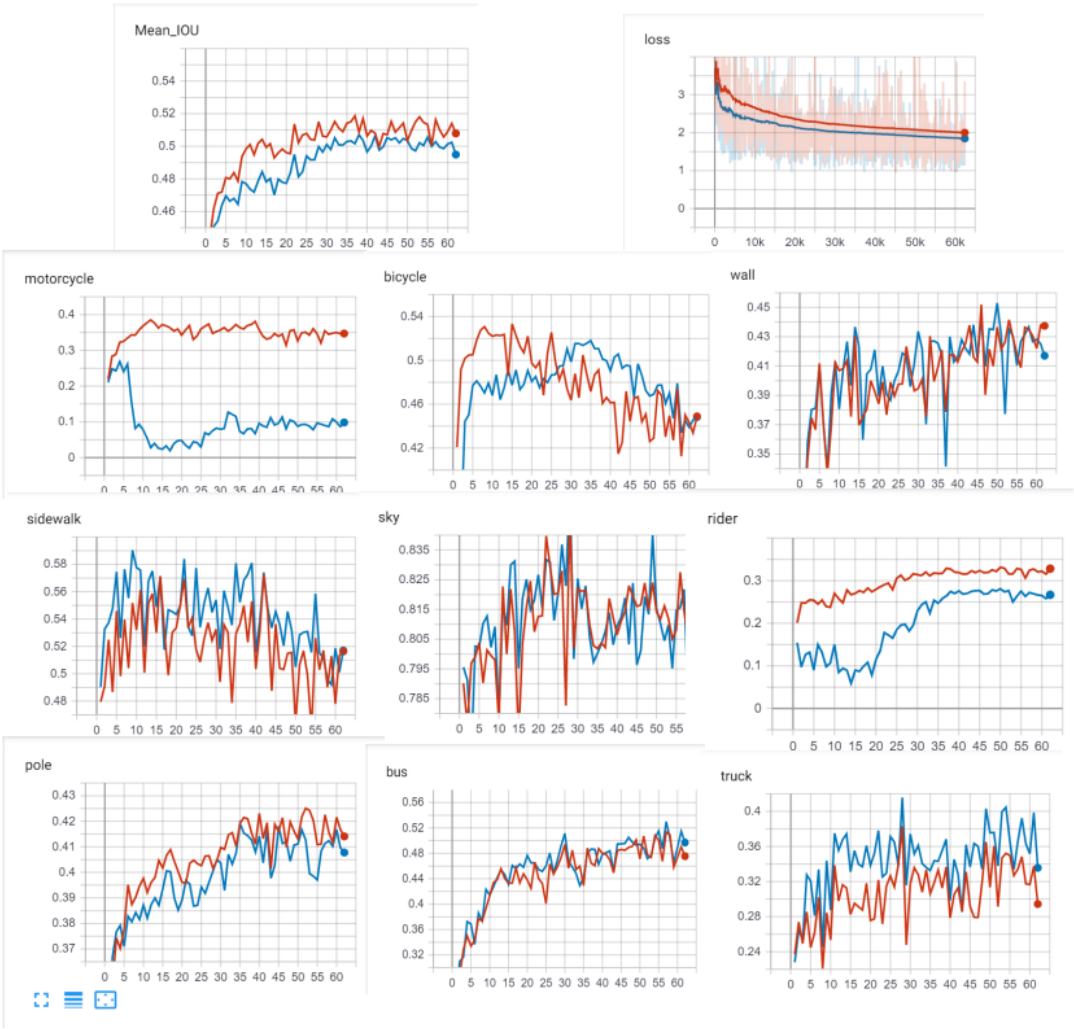

Figure 6.3: **Illustration of training curves of ProDA-PLG and ProDA-HR.** The brown curves represent the ProDA-PLG, and the blue curves represent ProDA-HR. The y-axis of the second picture is the overall loss of ProDA while those of other pictures are the absolute IOU values.

Figure 6.4: **Illustration of evaluation results of ProDA-HR with random cropping strategy and ProDA-HR trained with our pseudo-label-guided re-sampling strategy.** The orange boxes highlight the confusing area containing overlapped 'motor', 'bike' and 'rider' classes. The ProDA-HR cannot distinguish the difference between 'motor' and 'bike' while the ProDA-PLG can generate more accurate predictions compared to the former one. This figure is the same as Fig 6.1.

6.2.2 Experiment Results

In this section, we compare ProDA [58], ProDA-HR and ProDA-PLG. As shown in Table 6.1, ProDA-PLG can gain 15.3 mIOU improvement, 1.2 higher than that of ProDA-HR which does not utilize PLG re-sampling strategy. In terms of per-class IOU, ProDA gets the highest IOU in 9 categories. ProDA-HR and ProDA-PLG respectively gain the highest IOU on 4 and 6 classes. What's more, the PLG re-sampling method's ability to mine hard samples makes ProDA-PLG gain 28.9 IOU improvement in the 'motor' category compared to ProDA-HR. The improvements of per-class IOU and mIOU of the proposed PLG re-sampling strategy show that it can make the UDA model more powerful and robust compared to the random cropping strategy. Meanwhile, the training curves in Figure 6.3 show that the ProDA-PLG can also converge well and the final results of ProDA-PLG overweigh that of ProDA-HR.

In terms of qualitative analysis, this re-sampling strategy brings ProDA-PLG brilliant ability to

Model	GPUs	road	sideway	building	wall	fence	pole	light	sign	vege	terrace	sky	person	rider	car	truck	bus	train	motor	bike	mIOU	gain
Source	-	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6	+0.0
FADA	4	91.0	50.6	86.0	**43.4**	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	**38.0**	85.2	31.6	46.1	6.5	25.4	37.1	50.1	+13.5
SAC	4	90.4	53.9	**86.6**	42.4	27.3	**45.1**	**48.5**	42.7	**87.4**	40.1	86.1	67.5	29.7	**88.5**	**49.1**	**54.6**	9.8	26.6	45.3	**53.8**	**+17.2**
ProDA	4	91.6	51.8	83.1	41.8	35.7	40.1	44.1	43.4	87.1	**43.3**	79.6	66.5	31.6	86.8	40.8	53.2	0.0	**45.6**	**52.8**	53.6	+17.0
IAST	2	**93.8**	**57.8**	85.1	39.5	26.7	26.2	43.1	34.7	84.9	32.9	**88.0**	62.6	29.0	87.3	39.2	49.6	23.2	34.7	39.6	51.5	+14.9
ProDA-PLG	2	91.7	57.0	82.3	41.6	35.1	42.8	41.1	43.5	86.7	38.3	80.7	66.1	30.1	86.5	36.1	50.8	0.0	39.6	47.3	52.5	+15.9
CAG_UDA	1	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	+13.6
ProDA-HR	1	90.5	57.3	80.6	43.0	32.6	41.3	43.4	47.4	86.4	36.4	81.4	66.7	27.4	86.7	35.5	48.6	0.0	8.1	50.1	50.7	+14.1
ProDA-PLG	1	89.6	55.0	79.5	37.9	**38.3**	42.1	44.1	42.8	87.2	36.4	80.2	**67.0**	33.0	86.9	33.3	46.5	0.0	37.0	49	51.9	+15.3

Table 6.2: Comparison results of GTA5→Cityscapes adaptation in terms of mIoU (ProDA-PLG with different batch size 4 and 8). The best score for each column is highlighted.

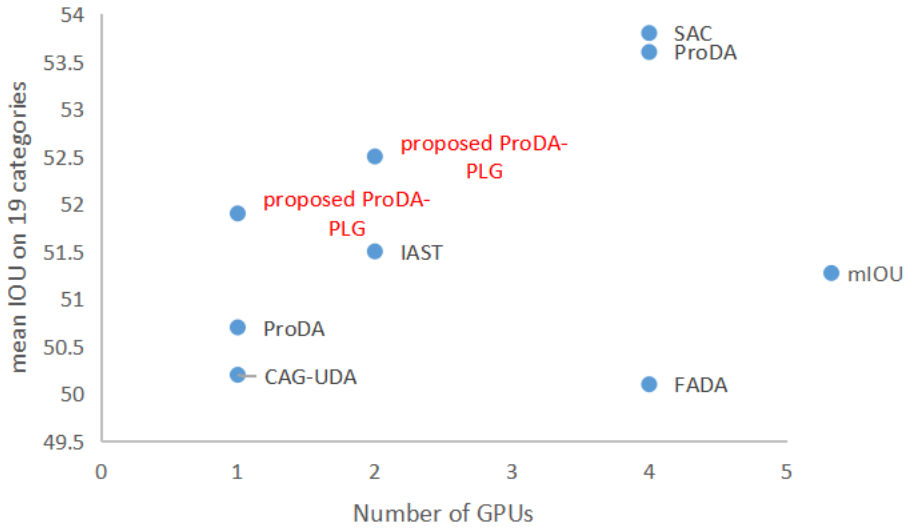

Figure 6.5: **Visualization of IOU scores and computational budgets for the proposed ProDA-PLG and other state-of-the-art methods.** The x-axis represents the number of used GPUs during training. The difference between the highlighted 'proposed ProDA-PLG' is the total batch size during training.

distinguish motorcycle and bicycle. As shown in Figure 6.4, the proposed re-sampling strategy solves the problem that ProDA-HR cannot distinguish the motorcycle from the bicycle, caused by the imbalanced distribution of 'motor' in the Cityscapes [13] dataset. In the predictions of ProDA-HR without PLG strategy, the pixels belonging to the 'motor' are misclassified as 'bike', and as for ProDA-PLG, the predictions are much more similar to the ground truth labels annotated by humans even if we cannot access them during training. This results from the wise sampling strategy compared to random cropping.

The mentioned ProDA-HR and ProDA-PLG can save three-quarters of computational budgets compared to the original ProDA while respectively getting only 2.9 and 1.7 mIOU drop. In order to prove proposed ProDA-PLG is robust and powerful, we conduct experiments when keeping 50% computational budgets compared to state-of-the-art IAST [35], FADA [51], SAC

[2], ProDA [58]. With more computational resources, ProDA-PLG can gain 52.5 mIOU in total which successfully decrease the IOU gap between ProDA trained with full computational budgets to only 1.1 mIOU while using half of GPUs to train (as shown in Table 6.2 and Fig 6.5).

6.3 Conclusion

In this chapter, we propose a pseudo-label-guided (PLG) re-sampling strategy to mine the small and rare objects in target domain. According to the quantitative and qualitative analysis of experiments, the proposed ProDA-PLG solves the problem of vanishing rare classes on target domain in Chapter 5. The IOU drops are restricted to 1.7 and 1.1 when given 25% and 50% computing resources and the ProDA-PLG significantly outweighs other state-of-the-art UDA methods when given the same computational budgets. By utilizing the ProDA-PLG to conduct experiments, researchers can get 200% or 400% of experimental efficiency compared to using the original ProDA [58]. It well answers the research question posed in this chapter.

Chapter 7

Conclusion

In this thesis, the aim is to train segmentation neural networks when the annotations are limited. We first raise research questions of whether the feedback mechanisms can improve the segmentation accuracy in full-supervised learning and semi-supervised learning. In Chapter 4, we introduce two different feedback mechanisms (MFB and ACFB) into the fully-supervised U-Net. We expect that iteratively feeding the generated masks to the input images can help the model suppress the background pixels and highlight the foreground ones. The experimental results show that simply concatenating the predicted masks and input images is meaningless. U-Net trained and evaluated with ACFB generate nearly the same masks as those predicted by the standard U-Net. And the multiplicative feedback mechanism can help the model suppress irrelevant background information and increase the IOU scores in some cases. But sometimes the partially wrong predictions at an earlier stage will lead to completely wrong masks at the final iteration. What's more, it is not a proper way to directly suppress the background pixels in the input images because global context might be lost in this manner. According to our experiments, directly feeding the output masks back to the input images cannot improve the segmentation accuracy in the fully-supervised approaches and of course, is not suitable for semi-supervised ones. In the future, the study of feedback mechanisms should focus on the feature level instead of the image level. Because in the feature space, the pixels belonging to the foreground objects in the feature maps have aggregated enough local and global context, which is helpful to the segmentation.

Then we focus on a similar research field to train the deep neural network with limited annotated data and unlabeled data in different distributions, unsupervised domain adaptation (UDA). It transfers the knowledge learned from a well-annotated synthetic domain to an unlabeled tar-

get domain. Although current self-training-based UDA frameworks gain significant accuracy, they require large computational budgets. So we raise the research question of whether self-training-based UDA frameworks can be trained with limited computational budgets, we successfully modify a state-of-the-art UDA method in Chapter 5. The proposed ProDA-HR saves 75% computational resources while keeping acceptable IOU scores. However, some rare and small categories vanish in the latent space of ProDA-HR.

In order to tackle the problem in Chapter 5, we raise a new research question of whether the vanishing classes can be recovered without extra computing resources and manpower. We answer this question in Chapter 6. By introducing a pseudo-label guided re-sampling strategy, our ProDA-PLG trained with 1 or 2 Tesla V100 can gain similar IOU scores to the original ProDA trained with 4 Tesla V100. In the future, we can introduce a dynamic updating scheme that updates the rare target categories in each iteration automatically instead of fixed ones which are determined before the training starts. This can make the proposed PLG strategy more flexible.

Bibliography

[1] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M Taha, and Vijayan K Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*, 2018.

[2] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15384–15394, 2021.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2019.

[5] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in neural information processing systems*, pages 5050–5060, 2019.

[6] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2956–2964, 2015.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[10] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[11] Jaehoon Choi, Taekyung Kim, and Changick 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*, pages 6830–6840, 2019.

[12] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[15] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[17] Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[18] Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham D. Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *British Machine Vision Conference*. BMVA Press, 2020.

[19] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[22] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[23] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *British Machine Vision Conference*, page 65. BMVA Press, 2018.

[24] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6728–6736, 2019.

[25] Jongmok Kim, Jooyoung Jang, and Hyunwoo Park. Structured consistency loss for semi-supervised semantic segmentation. *arXiv preprint arXiv:2001.04647*, 2020.

[26] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[27] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (Poster)*. OpenReview.net, 2017.

[28] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3367–3375, 2015.

[29] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, 2015.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[32] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

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

[34] M.Dimitrov. An investigation into feedback and representations in deep vision networks. 2020.

[35] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *European conference on computer vision*, pages 415–430. Springer, 2020.

[36] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[37] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[38] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[40] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio 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*, pages 3234–3243, 2016.

[41] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[42] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[43] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[44] Eisuke Shibuya and Kazuhiro Hotta. Feedback u-net for cell image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 974–975, 2020.

[45] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in neural information processing systems*, 2020.

[46] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[47] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[48] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[49] Hiroki Tsuda, Eisuke Shibuya, and Kazuhiro Hotta. Feedback attention for cell image segmentation. In *European Conference on Computer Vision*, pages 365–379. Springer, 2020.

[50] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[51] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *European Conference on Computer Vision*, pages 642–659. Springer, 2020.

[52] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[53] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[54] Y.Amano. Image segmentation based on narx architecture. 2020.

[55] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European Conference on Computer Vision*, volume 12351 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2020.

[56] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context network for scene parsing. *International Journal of Computer Vision*, 2021.

[57] Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1308–1317, 2017.

[58] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12414–12424, 2021.

[59] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 433–443, 2019.

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[61] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision*, 129(4):1106–1120, 2021.

[62] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.

[63] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[64] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.

[65] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. In *International Conference on Learning Representations*. OpenReview.net, 2021.