Unsupervised domain adaptation for object detection and whole slide image classification

by

Yuchen Yang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Deep neural network (DNN) has been developed rapidly in years. While it shows promising results in various tasks of computer vision, DNN typically suffers from accuracy loss due to the domain shift from a source domain to a target domain. To mitigate the accuracy loss without the label from target domain, unsupervised domain adaptation (UDA) approaches are proposed.

Compare to most UDA studies that target image classification and pixel-level classification (image segmentation), UDA for object detection is a relatively new area. A popular processing pipeline is to apply adversarial training with domain discriminator. The domain discriminator aligns the feature distributions of the source and target domain.

Existing methods in UDA object detection extract features from image level and directly adapt the full features as in UDA for classification tasks. However, alignment on full image level features as a whole is not ideal for object detection task. The presence of varied backgrounds could interfere with the result of adaptation. To avoid alignment on a full feature, this thesis proposes a novel foreground-focused domain adaptation (FFDA) framework. This FFDA framework mines the loss of the domain discriminators so that the alignment could concentrate on the foreground during backpropagation.

FFDA collects target predictions and source image labels and uses them to generate mining masks that outline foreground regions. And then it applies the masks to image and instance level domain discriminators to allow backpropagation only on mined regions. In addition, by reinforcing this foreground-

focused adaptation throughout multiple layers in the detector model, FFDA pushes the detector to gain a significant accuracy boost on target domain prediction. Compared with previous methods, FFDA method reaches the new state-of-the-art accuracy on adaptation from Cityscape to Foggy Cityscape dataset. The FFDA also demonstrates competitive results on other datasets that include various scenarios for autonomous driving applications.

In addition to object detection problem, this thesis also discusses the application of UDA for whole slide image (WSI) classification. Image classification for WSI is a challenging task compared to general image classification because of its high resolution and scattered key information. Previous work provided a novel deep Fisher vector coding pipeline for WSI classification. However, this pipeline suffers from the same accuracy drop phenomenon when deployed to another set of WSI from a different institution to perform the same task. This poses a limitation of the practical usage of the pipeline especially when the diagnoses of WSIs are hard to obtain.

On the other hand, previous works that apply UDA to medical imaging typically focused on adapting on small microscopy image samples or image patches extracted from WSI. UDA for the application of classifying the entire WSI has not yet been discussed due to the limited number of pipelines and datasets that support WSI classification.

This thesis aims at providing a UDA solution to enhance the robustness of the previous pipeline by mitigating the accuracy drop caused by different WSI datasets. This solution inserts the domain classifiers into the previous pipeline in different stages to align the features during training. The solution is evaluated by calculating confusion matrices before and after the adaptation. The results demonstrate that by placing domain classifiers in different stages the pipeline shows an accuracy boost on target WSI data.

Preface

Parts of the chapter 2 and 3 of this thesis have been published as Yuchen Yang, Nilanjan Ray, "Foreground-focused domain adaptation for object detection" in International Conference of Pattern Recognition (ICPR), 2020.

Chapter 4 and 5 of this thesis have been published as Yuchen Yang, Amir Akbarnejad, Nilanjan Ray, Gilbert Bigras, "Double adversarial domain adaptation for whole-slide-image classification" in The Medical Imaging with Deep Learning (MIDL), 2021.

All works are original.

Acknowledgements

My sincere gratitude towards my supervisor Prof. Nilanjan Ray for his years of supports and guidance. I very much appreciate that he gave me the exciting opportunity to explore the new frontiers in computer vision. The conversations with him always give me new inspirations. His encouragement pushes me forward when facing frustrations.

I am also grateful to Dr. Amir Hossein Hosseini Akbarnejad who has been patiently teaching me about his original library Pydmed and his work on whole slide image classification. His code and research set the keystone to mine. The discussions with him always teach me new knowledge in medical imaging.

Special thanks to Dr. Gilbert Bigras for providing the HER2 dataset for experiments and providing feedback to our research. And thanks to Dr. Seyed Moein Shakeri for providing valuable advice on my FFDA paper submission and rebuttal. Thanks also to Mr. Martin Humphreys for maintaining the server, as well as teaching me Gentoo and OS knowledge.

Finally, I would like to thank all my colleagues at the robotics & vision research group for being great listeners and providing feedback in group presentations.

Contents

List of Tables

List of Figures

Acronyms

CNN	Convolutional Neural Network
DNN	Deep Neural Network
DFVC	Deep Fisher vector coding
FFDA	Foreground-focused domain adaptation
FP	False Positive
GAN	Generative Adversarial Network
GRL	Grdient Reverse Layer
HER2	Human Epidermal growth factor Receptor 2
IHC	Immunohistochemistry
IOU	Intersection over union
KL	Kullback–Leibler
mAP	mean Average Precision
MIL	Multiple-instance Learning
MMD	Maximum Mean Discrepanc
NMS	Non-Maximum Suppression
RKHS	Reproducing Kernel Hilbert Space
ROI	Region of Interest
RPN	Region Proposal Network
SGD	Stochastic Gradient Decent
TP	True Positive
t-SNE	t-distributed Stochastic Neighbor Embedding
UDA	Unsupervised Domain Adaptation
WSI	Whole-slide-image

Chapter 1

Introduction

In this chapter, we describe the pervasive "domain gap" issue of DNN models deployed in real-world scenarios and the UDA approach to mitigate this issue. We introduce the particular challenges of using UDA in object detection and WSI classification, which are the main focus of this thesis.

1.1 Motivation

Deep neural networks (DNN) have been successfully applied and made to achieve human-level accuracy in disciplines across multiple computer vision tasks such as object recognition, segmentation and object detection [30].

Though amazing results are emerging from the utilization of DNN models, DNN models are still incomparable with the ability of human vision in the aspect of dealing with out-of-context objects. The performance of human vision is less affected by the surroundings, environmental changes, etc. It has been experimentally pointed out that simply adding random noise or changing the background will make the prediction of DNN less reliable [48]. To provide reliable prediction in different environments is crucial and challenging. Lack of adaptability will severely limit the applications of DNN models.

In research, the lack of adaptability is normally measured by using two or more datasets, i.e., when a network is trained on one or more datasets and applied on another to solve the same task, the model is unlikely to perform well. This phenomenon is caused by the data distribution difference between the two datasets, which is also referred to as the "domain gap."

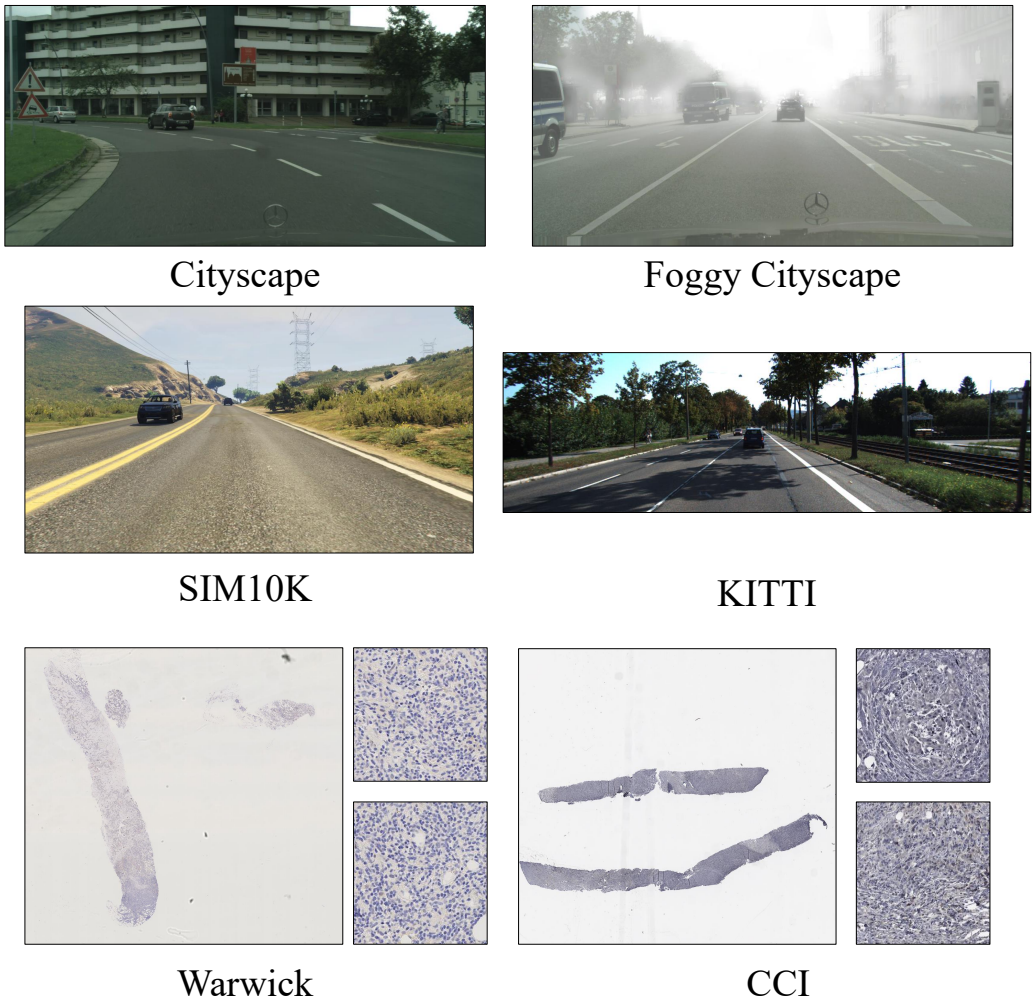

Figure 1.1: Illustration of the domain gap among datasets

The domain gap is pervasive in real-world applications, and it needs to be addressed with care. In past decades, the domain gap in various visual tasks have been observed and statically measured by reporting the performance differences of a model trained on a source dataset and tested on another dataset [58]. Such visual tasks include image classification, face recognition, object detection, semantic segmentation, person re-identification, image translation, image captioning, etc.

One of the representative real-world applications is autonomous driving. Past research has been focusing on the tasks of segmentation and object detection in autonomous driving which suffer from accuracy drop from one do-

main/dataset to another. An example of the visible domain differences is shown in the sample images of four datasets in the first and second rows in Figure 1.1. These four datasets represent various scenarios in autonomous driving where appearance changes happen and create obstacles for a model trained on one dataset and deployed on another. Top left image comes from Cityscape dataset [10] where the image are collected under clean weather. Top right image comes from Foggy Cityscape dataset [51] where the weather is foggy and the objects at a distance are hard to see. Middle right image comes from KITTI dataset [16] that applies a different resolution and varies in scales and categorical distribution. Studies aim to cope with these appearance changes that come from different weather/lighting conditions, different camera parameters, and background interference.

Learning to minimize the domain gap also inspires other applications. Industries and academia have been using synthetic images and simulators to train and test the developed algorithms. Synthetic images could be obtained from design engines such as CAD and Unity. Simulators (e.g. LGSVL [47]) are developed for a particular task to facilities the testing and development of AI software. The mid-left image in Figure 1.1 shows a sample from SIM10K dataset [24] which is collected by screenshots from the video game GTA. One of the applications for autonomous driving is to adapting from a synthetic dataset like SIM10K to a real-world dataset like Cityscape. The annotations of these synthetic data can be easily generated from a simulator, while the real-world data require massive effort in collecting and annotating by human labor. However, a model trained with synthetic data alone does not perform well on real data due to the limited generalization issues of the synthetic data. Minimizing the domain gap between synthetic and real-world data increase the accuracy in deployment and reduce cost for industries.

Besides the autonomous driving applications, the domain gap can also be found in whole slide images (WSIs). The bottom row in Figure 1.1 shows two example images. The left image is collected from the University of Warwick and the right image is from Alberta Cross Cancer Institution (CCI). Different chemical formulas used for staining, images scanned by different machines or in

different periods could all be part of the reason that causes the visual difference in the two images. The risk of misdiagnoses by the DNN model will increase if we can not provide a solution to mitigate the domain gap between the WSI datasets.

1.2 UDA background

A simple approach to solving the domain gap would be to apply supervised transfer learning, which brings in labeled data from the target dataset to the training set. However, labeling another dataset could cost a huge amount of effort, especially in the case of labelling WSI. Labels such as gleason scores of the WSI require extra specialists/equipment and therefore expensive to obtain.

Domain adaptation (DA) is considered as a sub-category [2] of transfer learning where the target annotations are not always available. There are several categories of domain adaption. The main focus has been on unsupervised domain adaptation. The unsupervised domain adaptation (UDA) approaches aim at reducing the accuracy drop without using annotated data from a new domain. In UDA, a model is trained on images from a source domain with labels, plus the images from the target domain without labels. Other categories of DA have also been developed. Few-shot DA [39] adapts to the target domain with few available annotations from the target dataset, multi-source DA [60] adapt from multiple source domains to target domain, partial DA [5] assumes that the categories in target dataset are a subset of the source dataset, etc.

This thesis focuses only on the UDA approach which works with one source dataset and one unlabeled target dataset. Other categories of DA typically are extensions of this UDA setting.

Early studies of UDA for DNN focus on classification task. Methods can be categorized as discrepancy-based, adversarial-based, and reconstruction-based [58]. Pioneering methods typically belong to discrepancy-based methods. These methods align the distribution by using statistical measurements

such as maximum mean discrepancy (MMD) [17] which is estimated by:

$$MMD^2 = sup_{\|\phi\|_H \leq 1} \|E_{x^s \sim s}[\phi(x^s)] - E_{x^t \sim s}[\phi(x^t)]\|_H^2 \tag{1.1}$$

where x is the image data. s and t represent source domain and target domain distributions, respectively. ϕ represents the kernel function that maps features to reproducing kernel Hilbert space (RKHS), $\|\phi\|_H \leq 1$ defines ϕ as set of functions in the unit ball of RKHS. Another popular measurement is correlation alignment (CORAL) [56], which is estimated as:

$$l_{CORAL} = \frac{1}{4d^2} \|C_S - C_T\|_F^2 \tag{1.2}$$

where C_S and C_T represent source and target feature covariance matrix, d is the dimension of the feature space. F represents the squared matrix Frobenius norm.

With the GAN [18] developed for image translation, adversarial-based methods that apply GAN loss - adversarial loss has become a popular and effective way to align feature globally. Early work by Ganin et al.[15] proposed a domain discriminator for aligning feature using adversarial loss. The domain discriminator is a small neural network that typically consists of few convolution layers. The domain discriminator accepts features extracted from a fully connected layer of a DNN model, and outputs two scores that try to discriminate whether the feature is extracted from the source or the target data. To align the extracted features from two domains in training stage, a min-max loss to fool the discriminator. This is achieved by attaching a gradient reverse layer (GRL) prior to the domain discriminator. The GRL will reverse the gradient that pass from a top layer to a bottom layer during the backpropagation. To put in a formal description:

$$\min_{\theta} \max_{w} L_d(\theta, w) \tag{1.3}$$

where w represents the weights of DNN up to the layer performing feature extraction. θ denotes the weights of the domain discriminator. L_d is a logistic loss on the two output scores. The goal is to update the θ so that the loss is

maximized, and update w so that the loss is minimized. The GRL to reverse the gradient $\frac{dL_d}{d\theta}$ to $-\lambda\frac{dL_d}{d\theta}$ (where $\lambda = 0.1$) during the backpropagation.

Though massive progress has been made, most of the approaches that specifically target classification task are not always directly applicable to other tasks. Due to the various nature of different computer vision tasks and the various pipelines in solving them, a universal DA pipeline may not be able to achieve optimal adaptation results for different tasks. Task-specific solutions are needed based on our observation.

1.3 UDA for object detection

1.3.1 Current techniques

Unlike the classification and segmentation (pixel-wise classification) tasks, it is only recently that research starts to focus on developing task-specific UDA solutions for object detection in minimizing the domain gap.

Object detection task has its unique characteristics and processing pipelines. The objective of object detection task is to both locate the position of the object and classify the object to a specific category. Objects in object detection task typically occupy less area on an image than other tasks. Pipelines of solving object detection task are commonly categorized as one-stage and two-stage. Techniques for UDA object detection are mainly focusing on two-stage detectors.

Current techniques to tackle UDA object detection can also be categorized as the same categories in the classification tasks (refer to Section 1.2).

We primarily focus on the pipeline of adversarial-based methods [8], [19], [50], [54], [59], [63], [69], which seek a joint space where the source and target features are aligned by fooling the domain discriminator. Compared to other pipelines, such as discrepancy-based and reconstruction-based that require extra training rounds/steps before or after training the detector, the methods under adversarial-based pipeline are typically trained once along with the detector and achieve high accuracy.

Most of the existing works on UDA object detection use Faster RCNN as

the baseline detector due to its high efficiency and robustness. Faster RCNN can be divided into two parts: the image feature level, where the full image area is processed, and the instance feature level, where each feature represents an object. Early method [8] extracts features from both levels and aligns them by applying a min-max adversarial loss for the image and the instance level domain discriminators. However, aligning full image features is not ideal for object detection. Compared to other tasks, such as object recognition, the foreground (i.e. object) areas, especially for autonomous vehicle applications, are relatively sparse on an image, while the background area is significant. A traditional domain discriminator will seek the hyperplane to separate and align the foreground as well as the background. Alignments of sophisticated layouts of structures and appearances of the background are likely to pose difficulties for minimizing the domain gap.

Most recent works [50], [54] adopt the focal loss [35] for domain discriminator and put a focus on adapting the distant features. He et al.[19] weighted the gradients from backpropagation in the adversarial part based on prediction scores.

Both focal loss and weighted gradient result in adapting the feature with certain preferences in local areas instead of adapting the feature equally on a full feature level. This preference, as shown in the experiments in paper [50], is visualized as the adaption tends to happen in the foreground area. However, these methods fail to exclude the background area directly. They implicitly cling to the assumption that background provides a positive influence in reducing the domain gap.

Zhu et al. [69] ignored image level adaptation to avoid full feature alignment and chose to cluster and adapt the area of instance level features in several windows with predefined sizes, which cover significant objects. Although the participation of background is reduced in this work, we argue the impact of adapting background is still ambiguous due to the less effective and coarse criterion they use in reducing background areas. We also posit that instance level alignment alone is not as effective in producing strong alignment of features.

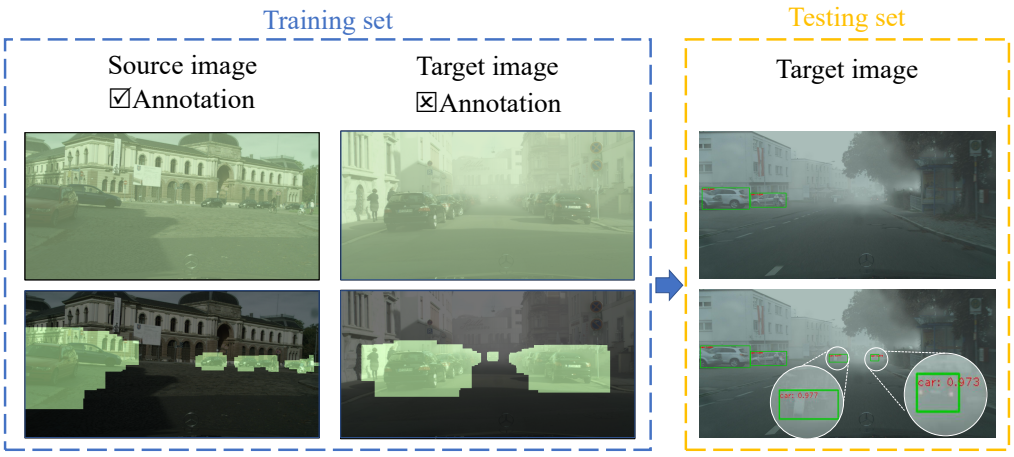

Figure 1.2: Illustration of the past methods vs. the proposed FFDA idea

1.3.2 Reducing background influence in UDA for object detection

As described in the above sections, though progress has been made to UDA object detection, it remains a question that whether the background information is posing a positive or negative impact on the adaptation result.

We hypothesize that, the existence of a large area of background from source and target images is still posing a significant and unpredictable influence on the domain adaptation, hence disrupting accuracy.

To test the aforementioned hypothesis, this thesis proposes a novel method for UDA object detection by guiding the domain discriminators to a foreground-focused adaptation. We exploit valuable information within each iteration of the detector training process for guiding the adaptation only on foreground regions. Note that in the previous adversarial-based methods, valuable information from both source labels and target predictions is not effectively utilized to aid the adversarial parts. The ground-truths in the source domain are mainly used only in calculating the detector loss, while the information it contains to distinguish between the foreground and background is ignored in the adversarial parts. The prediction scores for target images are also ignored in most adversarial training methods due to their instability during training iterations.

We propose to use source ground-truth and target predictions during training iterations to generate masks that mark the foreground areas for adaptation. Next, the pipeline applies the masks to mine the adversarial loss on both image and instance level domain discriminators to allow backpropagation only on the foreground areas. Furthermore, we seek to form a strong alignment of the foreground objects on different feature levels by placing the FFDA parts throughout the image feature level layers.

Example comparison of past methods vs. FFDA can be viewed in Figure 1.2. Green area indicate the area used for adaptation. Previous research (e.g.MLDA[59]) operate on full feature (top row) vs. the proposed FFDA (bottom row) that focuses on adaptation on foreground area. Right column is a sample prediction result on Cityscape to Foggy Cityscape from MLDA (top) and the proposed FFDA (bottom) on target domain. The two zoom circles show cars in distance that are detected by FFDA while failed by MLDA.

The key contribution of this thesis is to discover a new design to illustrate the importance of focusing on adapting the foreground areas in UDA object detection. The FFDA pipeline is straightforward and effective, as illustrated by experiments conducted across multiple datasets for autonomous driving. Compared to the previous methods in UDA object detection, FFDA reaches the new state-of-the-art accuracy of 40.1% mAP on Cityscape to Foggy Cityscape. In addition, compared to the previous adversarial-based methods, the FFDA achieved state-of-the-art accuracy of 25.6% mAP on BDD100K daytime to nighttime adaptation. It also reached 46.4% mAP on SIM10K to Cityscape datasets and 42.0% AP on KITTI to Cityscape dataset that are competitive with the current state-of-the-art methods.

1.4 UDA for WSI classification

Deep learning for classification on whole slide image (WSI) is a huge challenge. The challenge comes from its high resolution and the scattered diagnostic information. Recently, an embedding method with Fisher vector distribution encoding is proposed [1]. In this thesis, we further enhance the pipeline in the

aspect of coping with data distribution shift.

As discussed in Section 1.1, the domain gap in WSI datasets could be caused by various factors. Such factors include different staining processes by different institutions, WSI scanned in different periods and machines, etc. These factors result in accuracy drops that limit the practical deployment of a trained model for medical diagnosis. This thesis is dedicated to utilizing UDA approach to mitigate the accuracy drop between WSI datasets. UDA approach requires no annotation from the dataset to be deployed, therefore it could boost the accuracy with minimum cost for the generalization.

UDA for medical imaging has been discussed in several previous works [23], [29], [44], [57], [64]. However, the pipelines in these works only focus on adapting image patches extracted from the WSI to achieve higher patch-based accuracy. Their approaches are limited by the patch-based classification pipelines without consideration on adapting and classifying the WSI as a whole.

The recently developed deep Fisher vector coding pipeline (DFVC) [1] is an end-to-end pipeline that is capable of predicting a single label for the entire WSI. It generates two different feature representations for local patches and global information respectively. This design provides opportunity to consider the distribution shifts in both the patch level and the global level to jointly boost the accuracy. Established on this work, we proposed UDA solution integrates the pipeline with domain classifiers in two stages to minimize the accuracy decrease between datasets. We demonstrate the effectiveness of the proposed solution in HER2 breast tissue dataset. The experiments include the accuracy of a model without adaptation, adapted model, and oracle model to demonstrate the successful adaptation of our method.

1.5 Thesis statement

This thesis focuses on unsupervised domain adaptation solutions for object detection and whole slide image classification.

For object detection, contradict to the previous works that insist on adapting full feature, we hypothesize that domain adaptation on background poses

obstacles to aligning features, and excluding background area in UDA for object detection could benefit the adaptation result. To verify the hypothesis, we propose a straightforward and effective pipeline that mines the foreground area during adversarial training by gathering information from detection model. The evaluation of this pipeline is done by testing on multiple datasets related to autonomous applications and various ablation studies.

For whole slide image classification, we propose a novel UDA solution as an enhancement to the previous Fisher vector coding pipeline. This solution is the first one that considers the adaptation of WSI as a whole instead of extracted patches to minimize the influence of domain gap. Two stages of the previous pipeline are selected to place the domain classifiers to consider both local and global feature distribution shifts. The evaluation is done by testing the solution on HER2 data provided by a private dataset and online public dataset.

Chapter 2

UDA for object detection

In this chapter, we focus on the UDA approaches for object detection. We first introduce the classic one-stage and two-stage pipelines of DNN for object detection. Then, we discuss the current UDA pipelines for reducing the domain gap in object detection. After the establishment of previous works, we describe our proposed foreground-focused domain adaptation (FFDA) pipeline for object detection.

2.1 Preliminary and related works

2.1.1 One-stage and two-stage detectors

CNN-based object detectors can be mainly categorized into types: two-stage and one-stage. Two-stage detectors first extract regions of interest (ROIs), then predict the final object class and offset based on the region. A representative detector in this category is Faster RCNN [45]. The Faster RCNN introduced a sub-network named region proposal network (RPN). As shown in Figure 2.1, the RPN generates class-agnostic ROIs by using pre-defined anchors. There are k=9 anchors used in the original Faster RCNN network. By applying sliding windows on the output of backbone network, RPN generate 2k scores on each location and 4k coordinates. The 2k scores are the classification scores of k anchor boxes. The 4k coordinates are for the location regression, they represent the relative translation and scale that apply to an anchor with a particular size/ratio.

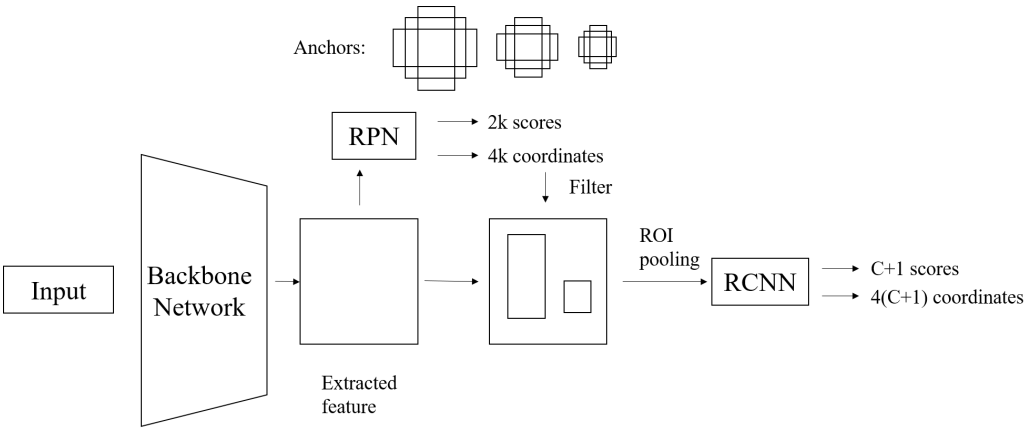

Figure 2.1: Faster RCNN illustration

The training loss for RPN, as described in paper [45], is estimated as:

$$L_{RPN}(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{2.1}$$

The p_i is the classification score for i-th proposal/anchor. p_i^* is the groundtruth label and is set to 1 if the anchor covers an object, 0 if the anchor is negative. t_i, t_i^* are the offset associated coordinates (relative offset and scale) of the prediction and groundtruth.

After a filter operation that eliminates the out-of-boundary predictions and applying non-maximum suppression, the RPN will select top 300 boxes for further processing. The ROI pooling is applied on the shared extracted feature guided by these boxes. The features are pooled by a 7x7 grid and flattened to fixed-sized 1D features. The final attached RCNN contains several fully connected layers and generates C+1 (C categories and 1 background category) scores for each ROI and 4(C+1) scores for location regression.

Classic one-stage detectors, such as SSD [37] and YOLO [43], usually use pre-defined anchors and directly predict the category and location. More recent methods [11], [34], [35], [65] made advanced modifications to boost accuracy while following the same footsteps of two-stage and one-stage pipelines.

In UDA object detection, most prior works [4], [8], [19], [21], [25], [27], [50], [53], [54], [59], [63], [69], [70] aim at improving domain adaptation of Faster RCNN. Thus, we choose Faster RCNN as our basic detector model for adaptation. Notice that our idea of the 'foreground-focused' adaptation is

straightforward and has no modification on the detector model itself during testing stage, and should be applicable to other detectors as well.

Because UDA for object detection is a relatively new direction, there is no systematic categorization on current UDA research. We follow the notion in the survey of Wang et al. [58] on the categories of UDA for classification task, and hereby categorize current methods to one of adversarial-based, reconstruction-based, and discrepancy-based.

2.1.2 Adversarial-based UDA pipeline

A number of recent methods [8], [19], [50], [54], [59], [63], [69], [70] adopt the adversarial training pipeline that attempts to align the intermediate feature representations from the source and target domains into a joint representation. This line of work typically attaches domain discriminators [15] to the detector to distinguish the source and target domain features. The domain discriminator is then deceived by applying an adversarial loss to achieve feature alignment.

A pioneering work [8] proposed DA Faster RCNN which applies two domain discriminators with gradient reverse layer (GRL) layers and placing them into the image level bottleneck layer and fully connected layer to achieve image and instance feature invariance. Their loss is estimated with four components as follow:

$$L = L_{det} + \lambda(L_{img} + L_{ins} + L_{cst}) \tag{2.2}$$

L_{det} is the loss of the original Faster RCNN detection loss. L_{img} and L_{ins} are the loss components of the image level adversarial loss and instance level adversarial loss. L_{cst} is a regularization that applies a $l2$ distance on the outputs of image level and instance level domain discriminator to provide consistency on domain predictions. λ balances the detection loss and the rest of the DA loss terms.

A multi-adversarial framework is proposed in works [19], [59], and later used by more advanced works [54], [70], to make multi-scale features invariance.

This invariance is achieved by extracting features from intermediate layers in the backbone and attach them with domain discriminators.

Xie et al. [59] proposed a multi-adversarial framework - MLDA that adapt image level features extracted from n=1 to 4 intermediate layers and the instance level feature. The selection of layers is based on a regular interval. They provide experiments that compare the multi-adversarial pipeline versus the DA Faster RCNN and showed a significant improvement.

He et al. [19] proposed a SRM unit for image level adaptation that downsized the features to boost training efficiency for the multiple layers adaptation. The SRM uses a 1x1 convolution layer to reduce the channel of extracted features and then realign the features to reduced scales with increasing channels. A weighted GRL is also introduced to adjust gradients from an instance level domain discriminator based on domain predictions.

$$G = -\lambda(d * p + (1 - d)(1 - p))G \tag{2.3}$$

d represents the domain label and p is the prediction of the domain. The gradient is reduced when the feature is less distinguishable to the domain discriminator. λ is a hyper parameter. Although SRM provides additional efficiency in training, the accuracy measured by average precision is inferior to MLDA. And weighted GRL only provided less than 1% improvements.

Zhuang et al. [70] applied a similar multi-adversarial pipeline on image level feature alignment. In addition, they explored the category information to establish strong alignment on instance level by constructing instance pairs based on the domain and the category information to train the instance level discriminator. They also backtrack the ROIs to image level feature and adding a correlation loss on the instance pairs, which further reduce the distance of features from different domains but in the same category and discriminate features in the different category from the same domain.

UDA for object detection typically adopt the same logistic loss for training a domain discriminator. Alternative loss terms and context vectors were investigated by Satio et al. [50] and Shen et al. [54].

Satio et al. [50] proposed SWDA. They attach an image level discriminator

with focal loss [35] to form a weak alignment on the bottleneck features of Faster RCNN, while features in the lower layers are strongly aligned. The focal loss is estimated as:

$$FL(p_d) = (1 - p_d)^{\gamma} * log(p_d) \tag{2.4}$$

p_d is the prediction for domain from the domain discriminator. $d \in \{s, t\}$ indicates the domain is either source or target. By using focal loss on image level adaptation, the domain classifier will focus more on hard-to-classify examples. while ignoring the easy ones. γ is a weight that applies to control the influence of hard examples. An additional context vector is also introduced in SWDA. A context vector is a concatenated vector on features extracted from discriminators and is attached to the RCNN features to stabilize the training process. Due to the plug-in context vector, the detector model itself is altered, which restricted the detectors that this pipeline is capable of applying to. Though it is not summarized explicitly in SWDA, the Grad-CAM [52] result of the gradient of domain discriminator from their experiments shows a tendency of focusing the adaptation on foreground regions.

Shen et al. [54] investigated various loss terms including traditional cross-entropy loss, lest square lost, and focal loss. By using the multi-adversarial pipeline with multiple domain discriminators, they experimentally found an optimal order to attach these loss terms to the discriminators. A gradient detach policy for context vector is also used to prevent the gradient from the vector to affect the backbone during training.

To avoid aligning irrelevant areas, Zhu et al. [69] choose not to align image level features and mine the area for local level adaptation. They proposed to group the ROIs from RPN by K-means, and place windows with fixed sizes and ratios on the clustered centers. The 1D corresponding features which are inside a window are reassigned to one group. The grouped features are then processed with adversarial training. Instead of using the domain discriminator, they chose a conventional GAN alike model for adversarial training. They apply two generators to reconstruct the corresponding area from the grouped features for source domain and target domain respectively, and then use two

discriminators to distinguish whether the reconstructed area is from its original domain or a fake one reconstructed from another domain.

2.1.3 Discrepancy-based UDA pipeline

Discrepancy-based methods fine-tune the detector to mitigate accuracy drop. Many methods [25], [26], [49], [63] apply a self-training technique by selecting the high-confidence target predictions as target domain pseudo labels. Then, these methods retrain or fine-tune the detector with generated pseudo labels to obtain better predictions. The detector gradually improves its accuracy after additional training.

Yu et al. [63] obtained the pseudo label from the DA Faster RCNN. In the next training round, they added residual blocks to the detector only for target domain images, and they train this DA Faster RCNN again with the labeled source data and target data with pseudo label.

Kim et al. [26] tackled with the one-stage object detector SSD. They proposed a weak self-training scheme. They first calculate a reliable score on ROI, which is a product of the IOU and the confidence score. And then they threshold the score and conduct hard negative mining to mine the positive examples and negative examples. To avoid bias in self-training, they choose the negative examples with lower loss. A background score regularization is also proposed to increase the distance of foreground and background target features by using an adversarial loss during the training. And finally, a self-training is deployed to gradually improve accuracy over training rounds.

Khodabandeh et al. [25] proposed three steps learning stages. In the first stage, a source trained detector is applied to the target domain to generate noisy labels. In the second stage, an image classifier is trained on the source domain using the groundtruth to test on target predicted area to refine the noisy category labels from the first stage. In stage three, they train the detector with pseudo labels they collected and refined in the second stage. In addition, they apply the Kullback-Leibler divergence to align the classification scores with the RCNN prediction scores.

RoyChowdhury et al. [49] targets at the object detection in videos. They

obtain the high-confidence predictions by using a tracker to provide temporal cues to augment the pseudo labels generated from a source domain trained model.

Cai et al. [4] introduced the Mean Teacher paradigm to UDA object detection task. First, they train a student model on the source domain data with the label. And they randomly argument a target image into two samples and fed it to a student model and a teacher model. To achieve the adaptation, the student model uses the ROIs generated from the teacher model, and optimized to maintain three consistencies between the predictions from the teacher and student model. Inter-graph consistency ensures the graph relations between the predictions from the teacher and student model are the same. Intra-graph consistency ensures the similarities of the same category of the prediction in student model. And finally, region-level consistency ensures region predictions are the same. Together, they make the feature invariant.

2.1.4 Reconstruction-based UDA pipeline

Reconstruction-based methods apply generative adversarial network (GAN) [18] to translate images from source to target. GAN is a popular model to produce synthetic, target-like images. Theoretically, these synthetic images should have the same data distribution as the data in the target domain, plus the benefit that these data come with annotations which are directly copied from the source domain. The reconstruction-based methods can then adopt the supervised learning on the detector with these images to achieve adaptation. Among all kinds of GAN models, CycleGAN [68] is the popular choice and is used in previous works [3], [22], [32]. The cycleGAN supports unpaired images for training and has robust performances across datasets. This makes the CycleGAN a suitable choice for the setting of UDA where the target images do not have labels.

The CycleGAN used in UDA learns two mapping functions F, G to map the source domain images X_s to target domain X_t and the target domain images X_t to source domain X_s. Two discriminators are designed for each direction of the translation to distinguish the fake images vs. the domain

images. The CycleGAN applies two cycle consistencies on the translated back images $G(F(X_s))$ with the X_s and $F(G(X_t))$ with X_t. The consistency is expressed by l_1 distance.

Arruda et al. [3] applies the CycleGAN to real-world daytime car detection data and translate them to nighttime images. The detector is trained with the generated nighttime images and the annotations from daytime data.

Lin et al. [32] proposed a cycle-structure consistency framework that utilizes GANs to translate images as well as segmenting the objects during the training. Their pipeline aims at the situation where the source domain dataset contains segmentation groundtruth.

Inoue et al. [22] proposed a pipeline that could be described in three steps. First, they train the detector on the source domain data. Second, they use the images translated from the source domain using CycleGAN to fine-tune the detector. Finally, a set of pseudo labels is generated by applying the detector to target domain images, and the detector is once again fine-tuned with the real target domain images and these pseudo labels.

Rodriguez et al. [46] proposed to use CycleGAN to translate source domain images to 3 target domains. And during the training stage of the detector, they apply l_1 distance to the features of the source images and the features of translated images. The trained detector is then used to generate pseudo labels for target domain images. And the detector is trained again on the target images with pseudo labels.

Some methods [21], [27], [53] incorporate the image translation of CycleGAN with adversarial training by aligning feature distributions for the real images with those for the synthetic images, during the training stage of the detector.

Shan et al. [53] used the translated images as the source domain images and adapt to the target images. The adaptation is done by treat training the detector with a standard domain discriminator attached at the bottleneck layer of the detector.

Kim et al. [27] proposed to add domain diversification when translating images using GAN. They achieve that by attaching a constraint loss term to the

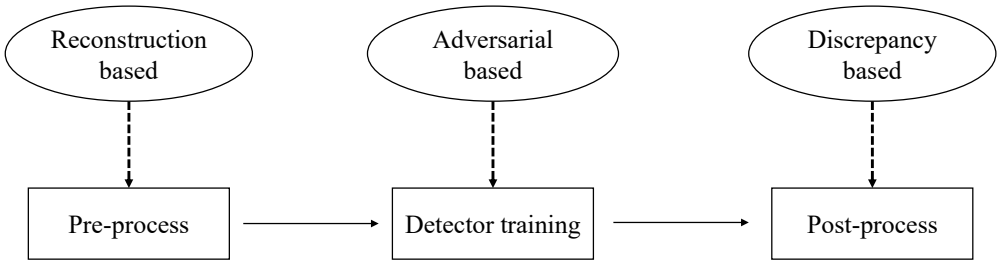

Figure 2.2: Pipelines for UDA object detection and the stages they apply to

GAN loss. They provided three constraint loss terms, one for color restriction purpose in which they calculate the l_1 distance between the generated images and the real target images, one for reconstruction purpose that is used in CycleGAN, and finally the combination of these two. They categorized the generated images to different domains based on the constraint loss term they use during the image translation. They apply an adversarial training pipeline using these images by attaching a domain discriminator that classify them to their corresponding domains.

Hsu et al. [21] treated the synthetic images as the intermediate domain. They apply the adversarial training on the detector using the source domain and the intermediate domain, then they conduct adversarial training again using intermediate domain and target domain. They also applied weight to the detection loss for the second adversarial training. This weight is extracted from the output of the discriminator in CycleGAN and its purpose is to minimize the influence of synthetic images which are not close to the target images during the training.

2.1.5 Summary

As we described in the above sections, we categorized current pipelines for UDA object detection into three categories: adversarial-based, reconstruction-based, and discrepancy-based. Different pipelines typically apply to different stages during the training procedures of a detector.

As shown in Figure 2.2, the solid lines link the stages in the proper or-

der to train a detector, dashed lines link the pipelines and the corresponding stages where the pipelines operate. Reconstruction-based methods generate synthetic images before training the detector. Adversarial-based methods are trained along with the detector. Fine-tuning and pseudo label generation in discrepancy-based methods are applied after one training round of the detector to reach stable prediction.

Adversarial-based methods typically yield better accuracy and they have lower cost in addition to the training of the detector. They do not rely on any prior or post processing steps but merely adding additional terms to the detector loss. However, most of the adversarial-based methods can only work with a specific detection pipeline, modifications are expected to adapt to other types of detectors. Reconstruction-based methods and discrepancy-based methods are independent to the detector since most of their steps happen before and after the detector training. But the training of GAN and extra training rounds lead to higher costs.

While pipelines vary greatly, different pipelines can potentially cooperate and jointly enhance the accuracy. This is because the pipelines are focusing on different stages during the training of a detector. In this thesis, we primarily focus on the adversarial-based pipeline for object detection.

2.2 FFDA for object detection

In this section, we demonstrate the proposed UDA pipeline that focuses on foreground adaptation in object detection.

We follow the UDA object detection setting, which is based on a source domain dataset with image data and groundtruth label: $D_s = [X_s, Y_s]$ and a target domain dataset with only image data available: $D_t = [X_t]$. To explain our method, we divide it into the following sections: First, we introduce the image level FFDA part that works on image level features. Second, we introduce the instance level FFDA part for the instance level features. And finally, we show how to integrate image and instance level FFDA parts with the detector in a multi-adversarial alignment.

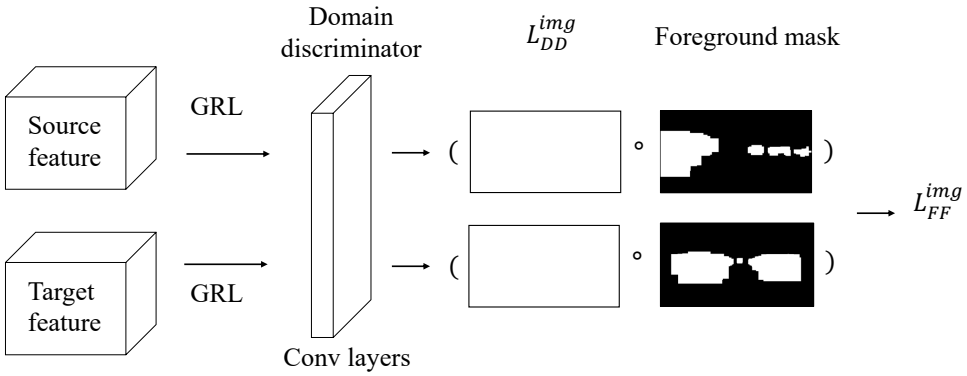

Figure 2.3: Image level adaptation illustration

2.2.1 Image level FFDA

Our essential idea is to limit adaptation only to the foreground areas. To achieve this goal, we propose to mine the loss in domain discriminator at the image level by using binary guidance masks for source images as well as target images. The masks should contain the information of the foreground locations to outline the regions where exactly the discriminator should adapt. To construct such masks for both source and target features, we choose to utilize the ROIs generated from the RPN model, the source domain groundtruth, and the final category predictions from Faster RCNN.

The RPN network generates class-agnostic ROIs, which contain the coordinates for potential foreground areas. For a traditional Faster RCNN training stage, it chooses a maximum of 128 foregrounds ROIs and randomly picks background ROIs to reach 256 ROIs in total to participate in the training. The foreground ROIs are identified by the IOU with the groundtruth.

We consider the ROI as a desirable unit for adaptation because it generates proposals and is used to make the final prediction. Our method extracts ROIs and utilizes foreground ROIs to form the masks. Our objective is to align only the areas inside the foreground ROIs and discard as many of the background areas as possible. A complete overview of image level adaptation is showed in 2.3. The dot between the loss and foreground mask is the element-wise product.

For source images, the foreground ROIs are easily obtained from Faster RCNN, because the ROIs are further used in the supervised learning to train the detector. To construct the mask, we set every pixel location inside a foreground ROI to one, while the rest of the pixel locations are set to zero on the binary mask. The examples of image level masks for source domain is showed in the top row of Figure 2.4. The experiment setting for this figure is to adapt from Cityscape (source dataset) to Foggy Cityscape (target dataset). Top row shows an example training image in Cityscape and the image level mask generated from its ground truth ROIs.

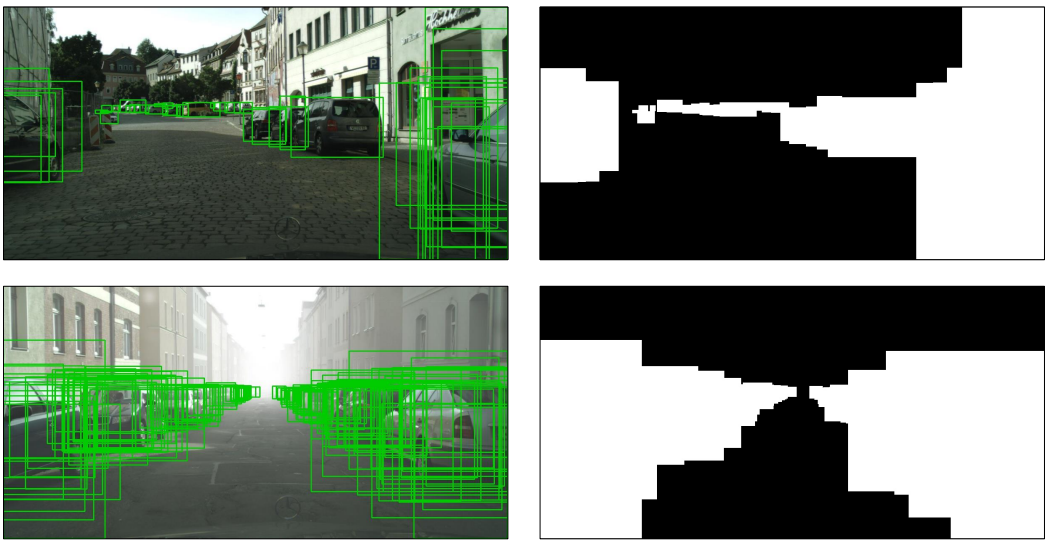

Figure 2.4: Image level masks for source domain

To construct a similar mask for target images, we borrow the final classification scores as a cue to outline the foreground areas. For every predicted ROI on a target image, the final prediction from Faster RCNN produces K category scores plus one score for the background.

We observed that the confidence scores for each foreground category are easily influenced by the unbalanced category in the training set. Thus, we avoid judging the foreground scores and instead choose to select the ROIs based on the background score as a descriptor of "to what extent an object belongs to the foreground categories." We experimentally discovered that placing a

relatively harsh threshold value T on target prediction is effective in providing foreground areas belonging to the target images. The ROI is considered a foreground object if the background score is lower than $T = 0.4$. Any pixels inside the eligible ROIs are assigned to one on the binary mask and zero otherwise. Bottom row in Figure 2.4 shows an example training image in Foggy Cityscape and the image level mask generated from its predicted ROIs.

During each training iteration, we obtain two aforementioned binary masks for the input source domain image and target domain image. We apply these masks to the domain discriminator loss to let the discriminator focus on the foreground areas we would like to adapt.

We employ a pixel-wise domain discriminator to cooperate with these masks. The structure of the domain discriminator in image level can be view in Table 2.1. The pixel-wise design provides space for avoiding alignment on the image level feature as a whole. If we wish to exclude the effect of certain regions on a 2D feature, we could ignore the corresponded pixels during backpropagation that represent this area or in another term - receptive field.

There are three image level domain discriminators are used and they share the same structure in this thesis. The three input/output sizes in Table 2.1 correspond the image level feature in three different layers, which is further discussed in Section 1.2.3.

The structure is the same as the previous works [59] which typically use two layers of Convolution with a Relu layer in between. This helps us to conduct fair comparisons with other methods, and role out the impact of the different structures of discriminators on the pipeline.

These discriminators provide 2D score maps in which every activation represents the score of a patch area on the input feature that belongs to the source or target domain. Before we apply the masks to the 2D score maps, we down-sample the masks to match with the 2D score maps for various sizes of input features. The 2D score maps can easily cooperate with the generated masks by only allowing the masked foreground areas to affect the weights in domain discriminator and the detector.

Since the masks are generated with the same size as the raw image data

Input feature, (h,w)=(75,150),(37,75),(37,75)
Gradient reverse layer, alpha=0.1
Convolution layer out_dim=512, kernel=3x3, stride=1, pad=0, bias=False
Relu layer
Convolution layer out_dim=2, kernel=3x3, stride=1, pad=0, bias=False
Output 2D score map

Table 2.1: Structure of global level domain discriminator

using labels and predictions, to cope with a specific domain discriminator placed in the network, we simply downsize the masks to fit with the size of the 2D loss map output from the domain discriminator.

We calculate cross-entropy loss for the discriminator in each location within the binary mask. The loss term for our foreground-focused patch-wise domain discriminator L_{FF}^{img} can be described as follows:

$$\begin{aligned} L_{FF}^{img} = & \frac{1}{N_s^{pixel}} \sum_{u,v} L_{DD}^{img}(u,v)_s M_s^{img}(u,v) \\ & + \frac{1}{N_t^{pixel}} \sum_{u,v} L_{DD}^{img}(u,v)_t M_t^{img}(u,v) \end{aligned} \tag{2.5}$$

where

$$L_{DD}^{img}(u,v)_{s,t} = -D_i \log(p_i(u,v)) - (1-D_i)\log(1-p_i(u,v)) \tag{2.6}$$

The notations N_s^{pixel} and N_t^{pixel} represent the number of "foreground" pixel locations in the down-sized binary masks M_s^{img} and M_t^{img}, respectively. $L_{DD}^{img}(u,v)_{s,t}$ calculates cross-entropy loss in the pixel location (u,v) in source and target features. D_i denotes the domain label, which is one if the i-th image is from the source domain and zero if it is from the target domain. $p_i(u,v)$ is the score output from the domain discriminator in the pixel location. With GRL attached ahead of the discriminator, we achieve feature alignment following adversarial training:

$$\min_{\theta} \max_{w} L_{FF}^{img} \tag{2.7}$$

where θ denotes the weights of the domain discriminator and w denotes the weights of the detector. The objective is to confuse the discriminator so that the features from the two domains are invariant to each other.

As we can observe from the loss term, only the cross-entropy loss in the masked area is propagated back to affect the weights in the domain discriminator and the detector.

2.2.2 Instance level FFDA

The image level FFDA is only conducting regional adaptation. Instance level adaptation is needed to align each feature of object as a whole. In the case of instance level where each ROI is represented by an individual feature, we directly use the foreground ROIs identified for image level FFDA to generate instance level mining masks M_s^{ins} and M_t^{ins}. Unlike the two-dimensional masks in image level, these instance level masks are one-dimensional vectors. The j-th value in the masks is set to one if the j-th ROI feature is considered a foreground ROI feature; otherwise, the value is set to zero. The pipeline for a instance level FFDA can be viewed in Figure 2.5. By using these masks, we can distinguish which ROI features belong to foreground and should be allowed to participate in the instance level adaptation.

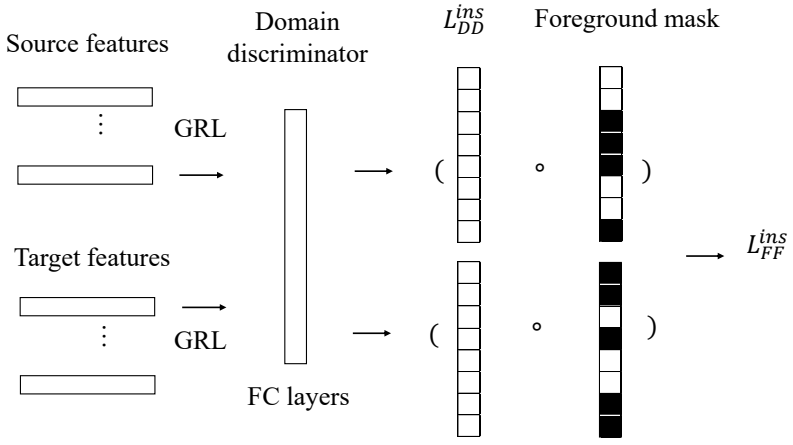

Figure 2.5: Instance level adaptation illustration

The structure of instance level domain discriminator follows the design of

Input feature, size=4096
Gradient reverse layer, alpha=0.1
Fully connected layer, out_dim=1024
Relu layer
Dropout layer, p=0.5
Fully connected layer, out_dim=1024
Relu layer
Dropout layer, p=0.5
Fully connected layer, out_dim=1
Output scalar

Table 2.2: Structure of instance level domain discriminator

Chen et al.[8], the details of the structure is listed in Table 2.2. This domain discriminator accepts a batch of ROI features, and output a single scalar for each feature. The scalar is one if the particular feature is from source domain, and zero if it is from target domain.

The instance discriminator is placed right before the final class and offset prediction on each ROI in Faster RCNN. The instance level FFDA loss L_{FF}^{ins} can be described as follow:

$$\begin{aligned} L_{FF}^{ins} = & \frac{1}{N_s^{feat}} \sum_j L_{DD}^{ins}(j)_s M_s^{ins}(j) \\ & + \frac{1}{N_t^{feat}} \sum_j L_{DD}^{ins}(j)_t M_t^{ins}(j) \end{aligned} \tag{2.8}$$

where

$$L_{DD}^{ins}(j)_{s,t} = -D_i \log(p_i(j)) - (1 - D_i) \log(1 - p_i(j)) \tag{2.9}$$

$p_i(j)$ is the output of the j-th ROI feature for the i-th image from the domain discriminator. The instance level discriminator loss $L_{DD}^{ins}(j)_{s,t}$ is the cross-entropy loss on the j-th ROI feature. N_s^{feat} and N_t^{feat} are the numbers of foreground ROI features for the input source image and target image, respectively. The instance level domain discriminator is also trained with GRL:

$$\min_{\theta} \max_{w} L_{FF}^{ins} \tag{2.10}$$

2.2.3 Multi-adversarial alignment

Previous works [19], [54], [59], [70] show that a hierarchical alignment in which multiple domain discriminators are placed across image level layers is effective in boosting adaptability. However, their attempts at boosting the image level feature alignment, as described above, are considered to hold back accuracy with sophisticated background interference. We adopt a similar multi-adversarial alignment concept with FFDA by directly placing multiple image level FFDA parts in multiple layers on the backbone of Faster RCNN. We show that consistency of applying image and instance level FFDA parts to the detector is necessary to reach the optimal accuracy across datasets. Details refer to Experiments section.

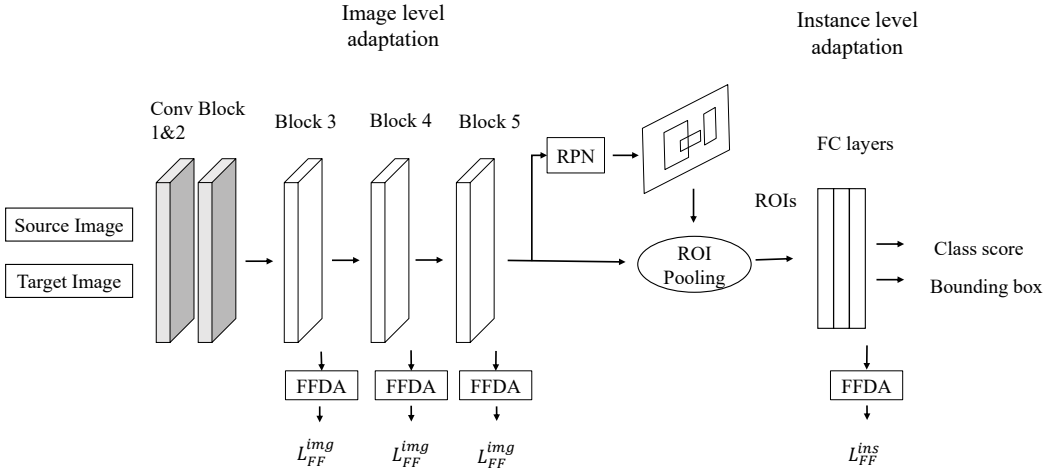

Figure 2.6: Multi-adversarial alignment illustration

Consider the Faster RCNN in this paper that uses VGG16 [55] as the backbone, the first two convolutions blocks are frozen during training and only the rest of 3 convolution blocks remain open for training. To directly observe the superiority of our FFDA, we place our discriminator after each of these convolution blocks (See Figure 2.6 Block 3,4,5) following previous works [19]. The overall loss term for our method can be summarized as follows:

$$Loss = L_{det} + \lambda(L_{FF}^{ins} + \sum_{k=3}^{5} L_{FF}^{img}(k)) \tag{2.11}$$

where L_{det} denotes the detector loss and $L_{FF}^{img}(k)$ is the image level FFDA loss after block k in the detector. λ is a hyper-parameter that balances the Faster RCNN detection loss against the adversarial adaptation loss. We set λ to 0.1 by experiment (refer to the ablation study section).

Chapter 3

Experiments of UDA for object detection

In this chapter, we evaluate our method on datasets for autonomous driving related applications. The content order of this chapter is as follow:

1. The implementation details of the FFDA pipeline including network and parameter settings.

2. The details of evaluation metrics that are used in the experiments.

3. Comparisons of our method and prior works on four sets of experiments by using the evaluation metrics.

4. The ablation studies for hyper-parameters, a detailed comparison on full feature alignment vs. foreground-focused alignment, visualization of feature alignment, and qualitative examples.

5. An alternative option for foreground identification: an attention-based method to generate the foreground mask.

6. Expansions of FFDA with discrepancy-based pipeline and reconstruction-based pipeline to observe the joint effect.

3.1 Implementation detail

We choose Faster RCNN with the VGG16 backbone network as our base detector. We follow the same Faster RCNN training procedures and hyper-parameter settings of prior works [45], [50], [54], [59], [70]. The whole network and the domain classifiers are jointly trained for 50K with a learning rate of

0.001 and 20K afterward with a learning rate of 0.0001 using stochastic gradient decent (SGD). Other parameters follow the same Faster RCNN setting. including 3 scales and 3 ratios for the anchors in region proposals generation (window sizes of 128x128, 256x256, 512x512 with ratios 2:1, 1:1, 1:2). The anchors with IOU larger than 0.7 is considered as foreground ROI for supervised learning. The final non-maximum suppression (NMS) is also set to 0.7 to avoid repeated bounding boxes in one location. The input is rescaled to match 600 pixels on the shorter side of image.

For all the experiments in section 1.3, hyper-parameter λ is set to 0.1 for balancing detection loss and adversarial loss, and the threshold for filtering the foreground target prediction T is set to 0.4. The experiments follow the UDA object detection setting which uses two datasets in every experiment: a source domain dataset that is fully labeled and a target domain dataset that does not have labels. The network is trained with a source image and a target image during one iteration. The UDA object detection accuracy is compared using the object detection evaluation metric–the mean Average Precision (mAP) -with IOU threshold of 0.5 on the target dataset.

In addition to comparing our work with previous works, we implement MLDA [59] and test it in every experiment. This implementation is directly built from our FFDA implementation by replacing the FFDA parts in our framework with full feature adaptation parts in MLDA. Therefore, we include the results for fair comparison and illustrate the effectiveness of our FFDA.

3.2 Evaluation metrics

In this section, we describe the several metrics used for evaluating the method including mean average precision, t-SNE and Grad-CAM.

3.2.1 Mean Average Precision

We adopt the evaluation metric widely used in measuring the performance of object detection algorithms, the mean Average Precision (mAP). The mAP is a metric that considers both precision and recall and the associated confidence

scores. To calculate the mAP, we first calculate the Precision and Recall scores for each category based on following equations:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

In object detection, a True Positive (TP) prediction must follow two conditions:

1) The Predicted bounding box must have an IOU that is larger than 0.5 with the bounding box in annotation.

2) The predicted category for the bounding box must be the same as the category of the overlapped bounding box in annotation.

If a predicted bounding box failed to achieve any one of the conditions, it is considered as a False Positive (FP) sample. And if the detector failed to predict a ground-truth bounding box, then this box is considered as a False Negative (FN) sample. The mAP is calculated differently in challenges such as COCO dataset [36] and Pascal VOC dataset [13]. In this thesis, we follow the implementation of the latter one which uses 11 point interpolation. The AP for each category in this metric is calculated as the surface under the Precision-Recall curve bases on 11 IOU partitions within a range of 0 to 1 (IOU from 0.0, 0.1 to 1.0). The mAP is finally calculated as the mean value of the AP scores for all categories.

For every set of experiment, we report 3 types of mAP scores:

1) Source trained mAP: The mAP score of the base detector trained on source dataset and tested on target dataset. This mAP score directly reflects the accuracy drop caused by domain gap.

2) Method mAP: The mAP score of the base detector trained with domain adaptation enhancement on source dataset and tested on target dataset.

3) Oracle mAP: The oracle mAP is estimated by training the base detector on target dataset and testing it on target dataset. This mAP score provides insight on the reachable accuracy when there is no domain gap.

3.2.2 T-SNE

The t-SNE [38] method is widely used for visualization of the feature alignment in domain adaptation. t-SNE is capable of dimensionality reduction on data, it projects high dimensional points such that they can be represented in a lower dimension. Similar high dimensional points will be represented by nearby points in the low dimension, while dissimilar high dimensional points will be represented by distant points.

To be more specific, the t-SNE first measures the similarity of data points represented by conditional probability $p_{j|i}$ by using:

$$p_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{3.3}$$

i and j are different indexes of data points, $p_{i|i}$ is set to 0, and σ_i is the Gaussian variance centered at point x_i. The joint distribution p_{ij} is calculated by:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{3.4}$$

The t-SNE defines a similar similarity measurement q which is applied to low dimension points $y_{i,j}$ as:

$$q_{i|j} = \frac{1 + \|y_i - y_j\|^2)}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)} \tag{3.5}$$

t-SNE minimizes the KL divergence C between p and q:

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{3.6}$$

The low dimensional points $y_{i,j}$ are updated in iterations by calculate the gradient $\frac{\delta C}{\delta y}$ and apply gradient descent to update y.

In this thesis, we apply the t-SNE to project the feature from high dimensions to two dimensions for visualization. The generated point cloud could represent the feature similarity, which is also a measurement of how good is the alignment of features from different domains.

3.2.3 Grad-CAM

Another metric we use for visualization is the Grad-CAM [52]. The Grad-CAM utilizes the gradients regards to the target prediction to generate a coarse

heatmap for the input image. The heatmap highlights the regions that are decisive to the prediction.

To be more specific, in the scenario of image classification, a target image is first fed to CNN to do a forward inference. Then, the task-specific loss - classification loss y^c is calculated by setting a selected target class c as the ground-truth. Through backpropagation, Grad-CAM captures the gradient flowed to the last convolution layer. The captured gradient can be described as $\frac{\delta y^c}{\delta A^k}$, with A^k represents the k-th activation map. By applying a global average pooling to the gradient, Grad-CAM obtains the importance weights α_k^c:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\delta y^c}{\delta A_{i,j}^k} \tag{3.7}$$

The final localization heatmap $L_{Grad-CAM}^c$ is generated by the combination of the importance weights and the activation maps with Relu:

$$L_{Grad-CAM}^c = Relu(\sum_k \alpha_k^c A^k) \tag{3.8}$$

Notice that the original paper applies the Grad-CAM only in image classification, caption and vision question answer tasks. Satio et al.[50] apply the Grad-CAM to understand the domain decision made by the discriminator they used. In this thesis, we also apply the Grad-CAM to visualize the gradient active regions reflected on the raw image.

3.3 Experimental results

We evaluate our method on four datasets for different scenarios in autonomous driving applications. We compare our method with prior works on the following datasets:

1) Clear to foggy weather adaptation (Cityscape to Foggy Cityscape),

2) Synthetic to real adaptation (SIM10K to Cityscape),

3) Cross-camera adaptation (KITTI to Cityscape),

4) Daytime to nighttime adaptation (BDD100k daytime to nighttime).

Methods	backbone	AD	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
Source trained	vgg16	✗	24.2	29.5	31.4	10.1	14.3	9.1	13.4	27.7	20.0
Diversify & match [27]	vgg16	✗	30.8	40.5	44.3	27.2	38.4	34.5	28.4	32.2	34.6
MTOR [4]	ResNet50	✗	30.6	41.4	44.0	21.9	38.6	40.6	28.3	35.6	35.1
RLDA [25]	Incep.V2	✗	35.1	42.2	49.2	30.1	45.3	27.0	26.9	36.0	36.5
PDA [21]	vgg16	✗	**36.0**	45.5	**54.4**	24.3	44.1	25.8	29.1	35.9	36.9
DA Faster [8]	vgg16	✓	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
Strong-Weak [50]	vgg16	✓	29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
SCDA [69]	vgg16	✓	33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
MAF [19]	vgg16	✓	28.2	39.5	43.9	23.8	39.9	33.3	29.2	33.9	34.0
SCL [54]	vgg16	✓	31.6	44.0	44.8	**30.4**	41.8	**40.7**	33.6	36.2	37.9
iFAN [70]	vgg16	✓	32.6	40.0	48.5	27.9	45.5	31.7	22.8	33.0	35.3
MLDA [59]	vgg16	✓	33.2	44.2	44.8	28.2	41.8	28.7	30.5	36.5	36.1
MLDA (Ours impl.)	vgg16	✓	32.7	44.6	44.9	24.0	40.4	34.1	30.5	34.9	35.8
Ours (block3,4,5+ins)	vgg16	✓	33.8	**48.3**	50.7	26.6	**49.2**	39.4	**35.8**	**36.8**	**40.1**
Oracle	vgg16	✗	36.2	45.8	52.7	33.4	51.5	44.0	37.8	39.0	42.6

Table 3.1: Experimental results on Cityscape to Foggy Cityscape adaptation

3.3.1 Normal to foggy weather

We test our method by training on the Cityscape dataset and adapt to Foggy Cityscape. The Cityscape dataset [10] collects urban street scene images from 27 cities. The Foggy Cityscape dataset [51] is constructed by rendering the Cityscape with depth-map-aided optical fog modeling. We follow the literature [8] in generating the rectangular bounding boxes from the original pixel-wise annotations as well as the training/validation set separation. There are 2,975 training images and 500 validation images labeled with 8 categories for both datasets. We compare our methods with previous works in UDA object detection. The mAP is reported on validation set of Foggy Cityscape.

The result is shown in Table 3.1, AP for 8 categories and mAP are reported. "AD" indicates if the method is an adversarial-based method. Oracle is a Faster RCNN trained with labeled target dataset. We note that the mAP for our method exceeds that of all the prior works that helped us to establish the new state-of-the-art accuracy in several categories. Compared to other adversarial-based methods, our methods provide the highest AP for six categories including person, rider, car, bus, motorcycle, and bicycle. Our method outperforms a Faster RCNN trained on the source dataset with 20.1% higher mAP. Compared to the MLDA which applies alignments on full features, our

method has 4.3% higher mAP by using the FFDA, which only focuses on adapting the foreground regions.

Notice that the SCL proposed by Shen et al. [54] conducts a thorough study on the effect of using diverse loss functions in multi-level alignment. And they expand the instance level feature of the detector with context vector as Satio et al. [50]. In contrast, our method employs cross-entropy loss in all domain discriminators and yields 2.2% higher mAP without modification on the detector, which indicates that our foreground-focused scheme is simple and effective.

3.3.2 Synthetic to real

Learning from synthetic data is helpful in relieving the effort of massive labeling. We use the adaptation result from SIM10K [24] to Cityscape to illustrate the superiority of our method in adapting from synthetic to real data compared to other adversarial-based methods. The SIM10K is composed of 10,000 images with labels captured from the video game Grand Theft Auto. We follow the experimental setting of SIM10K from previous papers and perform car detection. Car AP in Cityscape is reported in Table 3.2. Note that the Strong-Weak [50] does not provide AP for K to C. We train their method using their official code and report the score in the table.

Our method reaches the accuracy of 46.4% mAP which is close to the state-of-the-art result of 46.9% mAP from iFAN [70]. However, iFAN performs worse in the above experiment of Cityscape to Foggy Cityscape adaptation (4.8% lower than ours), while our method achieves a strong accuracy in both the scenarios. Similar to the result in Table 3.1, a clear accuracy difference can also be observed between MLDA and our method in synthetic to real adaptation. Multi-adversarial alignment on the full feature (MLDA) can boost AP by 2.8% compared with the DA Faster RCNN. With the substitution of our FFDA parts, the detector can boost 4.6% more mAP. These results indicate our success with adapting synthetic data to real data as the FFDA learns only foreground adaptation with less simulated background involved.

Methods	S to C (AP)
Source trained	34.9
DA Faster [8]	39.0
Strong-Weak [50]	40.1
SCDA [69]	43.0
MAF [19]	41.1
SCL [54]	42.6
iFAN [70]	**46.9**
MLDA [59]	42.0
MLDA (Ours impl.)	41.8
Ours (block3,4,5+ins)	46.4
Oracle	59.1

Table 3.2: Experimental results on SIM10K to Cityscape

3.3.3 Cross-camera

Cross-camera detection is another challenge for UDA object detection. The difference in the parameter settings across cameras affects the appearance of objects. We utilize the KITTI dataset [16] as the source training set and the Cityscape training set as the target training set. The number of training images in the KITTI dataset is 7,481. Unlike Cityscape which has a resolution of 2048x1024, the resolution in KITTI is 1224x370. The extreme scale difference from the resolution is an obstacle for object detectors to generalize from one to another. Cars in the KITTI dataset have different scales and ratios. Moreover, Cityscape is collected majorly in urban areas where the streets are surrounded by buildings while KITTI is collected in rural areas and highways.

The AP of car detection from KITTI (K) to Cityscape (C) adaptation is reported in Table 3.3. The result shows that a multi-adversarial technique alone (MLDA) only shows a limited boost of 1.4% AP in cross-camera detection compared to a source trained detector, but with full feature adaptation parts replaced by our image and instance level FFDA, the accuracy boost is 4.1% more in AP. And in general comparisons with other methods, our method reaches the second place while only have 0.5% difference with the state-of-the-art result from SCDA [69].

Methods	K to C (AP)
Source trained	36.5
DA Faster [8]	38.5
Strong-Weak [50]	37.9
SCDA [69]	**42.5**
MAF [19]	41.0
SCL [54]	41.9
MLDA (Ours impl.)	37.9
Ours (block3,4,5+ins)	42.0
Oracle	59.1

Table 3.3: Experimental results on KITTI to Cityscape

3.3.4 From daytime to night time vision

We evaluate our method on the BDD100k dataset on adaptation from daytime to nighttime domain. BDD100k provides 70K images for training and 10K for validation. In this paper, we focus on a challenging task which observes the accuracy on the adaption from daytime to nighttime. BDD100k provides 36,728 "daytime" images for training and 5,258 for validation, and 27,971 "nighttime" images for training and 3,929 for validation.

We compare our method on BDD100K with official implementations of DA Faster RCNN [8], Strong-Weak [50] and SCL [54] in Table 3.4, AP scores for 10 categories and mAP are reported. Notice that the nighttime environment itself is a challenging problem, even with with supervised training on an oracle model, the accuracy can only achieve 27.4% mAP. Our method achieves a 2.5% boost in mAP over the source-trained model with the highest AP across 6 categories, and our method has an average of 1% mAP advantage over other methods.

3.4 Ablation study

3.4.1 Hyper-parameters study

Our method has two hyper-parameters in addition to Faster RCNN: 1) Threshold T for filtering the prediction on target images to provide reliable foreground areas on target images. 2) Parameter λ, which is utilized to balance between

Methods	bike	bus	car	motor	person	rider	light	sign	train	truck	mAP
Source trained	20.2	33.6	45.7	12.1	27.6	14.0	16.1	31.0	0	30.3	23.1
DA Faster [8]	20.5	33.4	46.8	19.8	27.0	16.8	13.9	30.9	0	30.9	24.0
Strong-Weak [50]	19.6	33.0	46.5	**19.9**	26.4	18.6	15.6	31.5	0	30.9	24.2
SCL [54]	17.6	32.0	45.8	16.8	27.4	18.8	15.6	32.9	0	30.2	23.7
MLDA(Our impl.)	20.2	31.8	45.9	16.6	**27.7**	18.2	**16.9**	33.9	0	32.3	24.4
Ours(block3,4,5+ins)	**22.3**	**34.0**	**47.4**	19.7	27.4	**23.0**	14.6	**34.7**	0	**32.7**	**25.6**
Oracle	19.4	39.6	56.1	17.8	29.5	10.9	23	38.9	0	39.1	27.4

Table 3.4: Experiment results on BDD100K daytime to nighttime.

the detector loss and adversarial loss. We conduct a sensitivity study to these two parameters on two sets of experiments: Cityscape to Foggy Cityscape and SIM10K to Cityscape.

For threshold T, we increase the value from 0.1 to 0.6 with step size of 0.1. A larger threshold identifies more ROIs as foreground ROIs. As we can see from the Figure 3.1, a significant accuracy drop only happens after setting a threshold larger than 0.5 for SIM10K to Cityscape dataset. This is expected since a larger threshold will introduce more potential background areas from target image to adapt with the foreground of source image. Based on these experiment results, we choose to set T is to 0.4 for the majority of experiments.

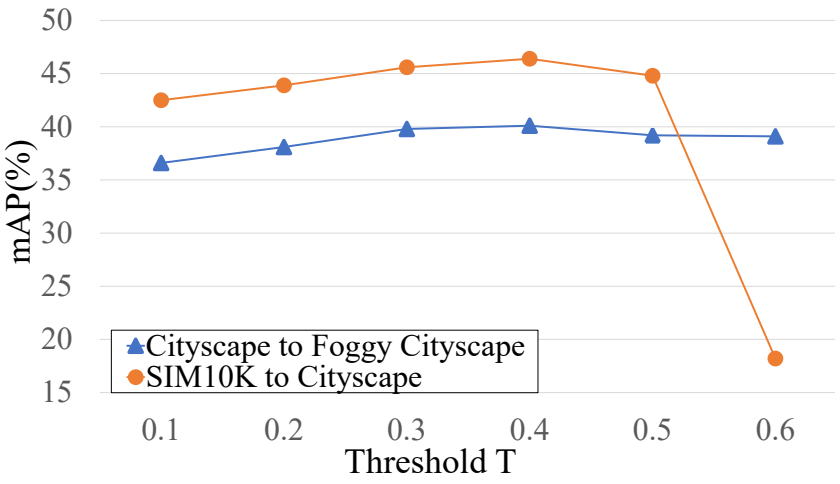

Figure 3.1: Sensitivity study on parameter T.

For balancing parameter λ, we increase the value from 0.01 to 0.25 with step size of 0.05. A smaller value for λ will make the impact of adversarial loss small, while a larger value will increase the significance of the adversarial

parts. According to the Figure 3.2, we can see that if λ is too small, the mAP is significantly lower compared to the peak value. And increasing the λ from 0.1 is not showing a significant change on mAP. Therefore, we choose to set λ to 0.1 for the majority of experiments.

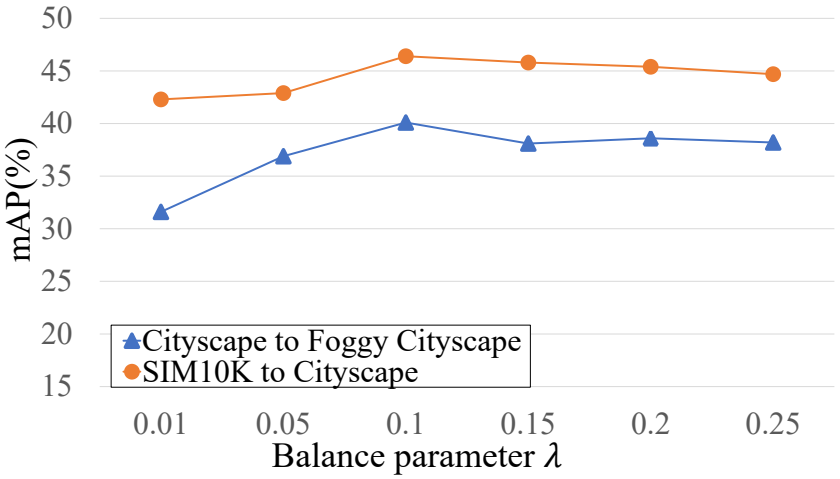

Figure 3.2: Sensitivity study on parameter λ.

3.4.2 Full feature alignment vs. foreground-focused alignment

As we described in the above chapter, the FFDA pipeline includes three image level domain discriminators and one local level domain discriminator, the experiments above are treating the pipeline as a whole. In this section, we first take a closer look at how do they work together to achieve better accuracy, then we compare the full feature adaptation and the foreground adaptation at various levels. We summarize the methods and statistics in this section in Table 3.5. The mAP in Cityscape (C) to Foggy Cityscape (F) and AP for car in SIM10K (S) to Cityscape (C) are reported. The MLDA reported in this table is from our implementation.

The FFDA and MLDA both use a multi-adversarial alignment that aligns features from block 3,4 5 on image level and features on instance level. We show the results of gradually increasing the number of layers for adaptation. The difference is that the adaptation in MLDA focuses on the full feature, while ours focus on foreground-focused regions. In table 3.5, methods without

block3 and block4 adaptation follow the design of DA Faster RCNN except the loss regularization. In this setting, MLDA has a slight advantage on mAP score in C to F experiment. However, this advantage varnishes after the block4 is added for adaptation in both methods. The accuracy boost of FFDA exceeds the MLDA as the number of layers for adaptation increases.

To test the influence of bringing in background adaptation, we replace the FFDA inside our framework with the domain adaptation parts that operate on the full feature in different levels in MLDA [59]. As we can observe from Table 3.5, in C to F experiment, replacing the instance level and block 4 domain discriminator to full feature alignment can cause 3% mAP loss. And for block 3 and 5, the replacement causes 1% mAP loss. In S to C experiment, replacing individual FFDA on image level to full feature alignment can cause an average of 2% mAP loss. This demonstrates that any attempts to operate on adapting full feature result in degeneration on accuracy. The best performance is achieved by having FFDA on every location consistently - blocks 3, 4, 5 in image level and instance level. The last row in Table 3.5 shows the result of a model which only adapts the background regions. The masks in this model are $1 - M_{s,t}$. We can see the adaptation on background regions reduce the accuracy in S to C experiment.

The experiments in this section illustrate the advantage of foreground-focused alignment over full feature alignment. And the high accuracy of a foreground-focused adaptation is based on a joint effort of the individual FFDA discriminator, multi-adversarial alignment, and a consistent attempt to enforce the foreground adaptation.

3.4.3 Visualization on feature alignment and qualitative results:

We apply t-SNE [38] on instance level features to observe the feature alignment results visually. We use the validation sets in Cityscape dataset and Foggy Cityscape dataset. In previous works, the t-SNE is applied to the feature extracted from the last layer of the backbone network. The result from their work could visually explain the feature alignment on the full feature. However,

Methods	C to F (mAP)	S to C (AP)
Source trained	20.0	34.9
DA Faster [8]	27.6	39.0
MLDA w/o block3 and block4 full feature	32.7	38.2
MLDA w/o block3 full feature	34.6	39.6
MLDA	35.8	41.8
Ours w/ full feature DA on instance level	37.5	45.5
Ours w/ full feature DA on image level block5	39.0	44.5
Ours w/ full feature DA on image level block4	37.2	44.2
Ours w/ full feature DA on image level block3	38.7	44.8
Ours w/o block3 and block4 FFDA	31.7	39.8
Ours w/o block3 FFDA	38.2	43.8
Ours	40.1	46.4
background-only adaptation	24.2	30.3

Table 3.5: Ablation study on individual component

it is not applicable when our focus is partial regions on the full feature.

To fairly compare only the foreground regions of the feature, we first extract the feature from block5 in the backbone, and then we apply ROI pooling to the ground-truth of objects. By doing this step, we unify the locations of ROI features in different methods. The pooled features are then further processed by t-SNE to provide two dimensional point clouds. This adjustment is a compromised solution, it helps to regularize the feature regions with different scales/ratios, but it also loses the spatial relations on the feature. The result is shown in Figure 3.3. Blue points represent target domain features, and red points indicate source domain features. The observation is that the ROI features are aligned better in our method with less obvious unaligned points (less separated blue and red area in our method).

We also provide the Grad-CAM result on Cityscape to Foggy Cityscape to see the difference of decisive regions between full feature alignment MLDA and our FFDA. For each image in the dataset, we first allow a forward pass to reach the end of the Faster RCNN. Then we calculate a loss which is the sum of the predicted classification scores of all foreground ROI features. The confidence threshold is set to 0.5 in this experiment. After backpropagation, the gradient is extracted from the last layer of the backbone layer to generate

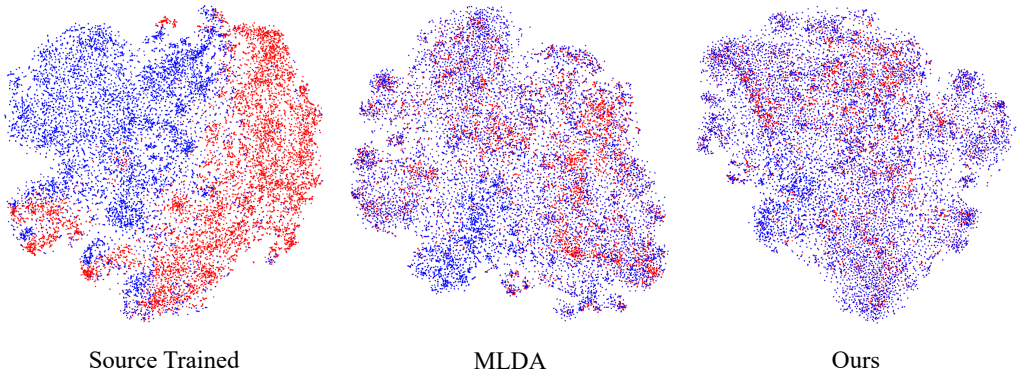

Figure 3.3: Visualization of instance level features using t-SNE[38].

the Grad-CAM result. As we can observe in Figure 3.4, our method that aligns foreground regions provides a higher response in reasonable foreground regions, while the heatmaps of MLDA are noisier and show lower response in the foreground regions.

Qualitative results are shown in Figure 3.5. The images are from the experiments of adapting from Cityscape to Foggy Cityscape and SIM10K to Cityscape. Only the predictions that have a confidence score over 0.5 are shown in the figure. It is clear to observe that our method is capable of detecting more objects in target domain after adaptation. Please refer to the Appendix for more results.

3.5 Alternative foreground identification

The FFDA pipeline utilizes foreground ROIs generated from the ground-truths to identify the regions for adaptation. Because the ROIs are measured using rectangle bounding boxes, it can be argued that the foreground regions are not precise on pixel level. Certain regions inside an identified foreground bounding box could be part of the background. Although we can not make the assumption that precise pixel-wise annotations are always obtainable for object detection task, we investigate a possible approach that obtains a pixel-wise score map from the DNN that depicts the ROI without the restriction of the rectangle shape.

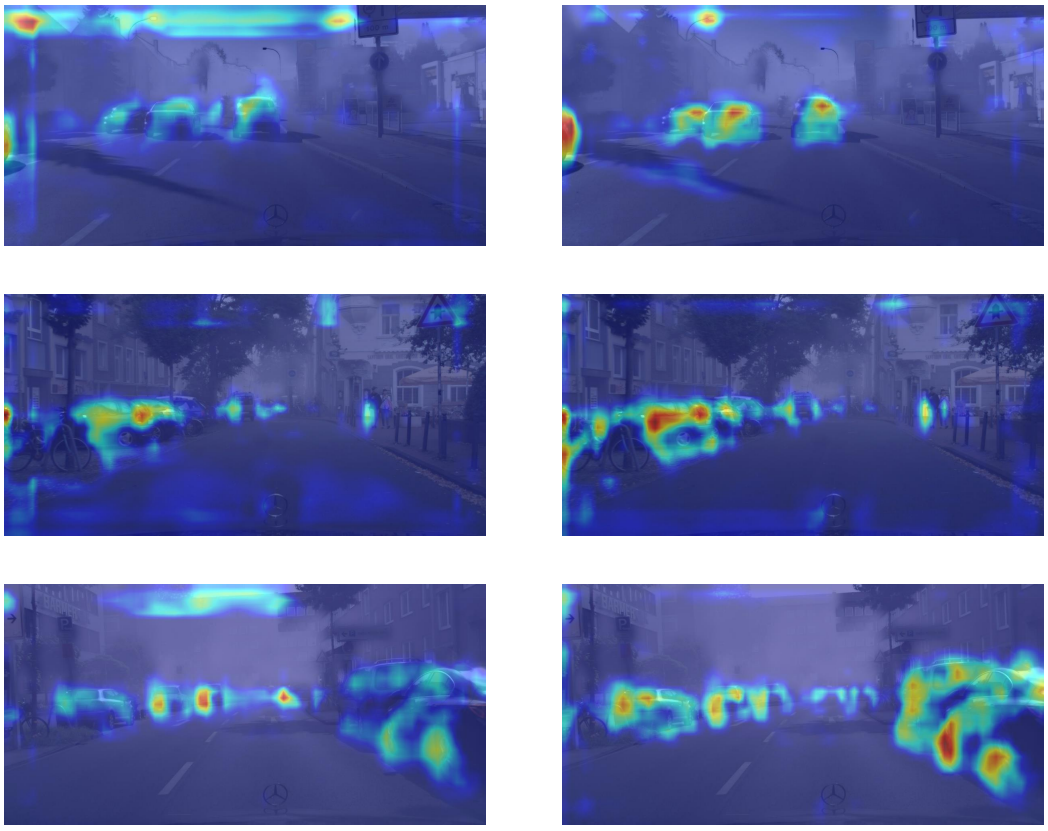

Figure 3.4: Grad-CAM results of MLDA and ours

Figure 3.5: Qualitative examples

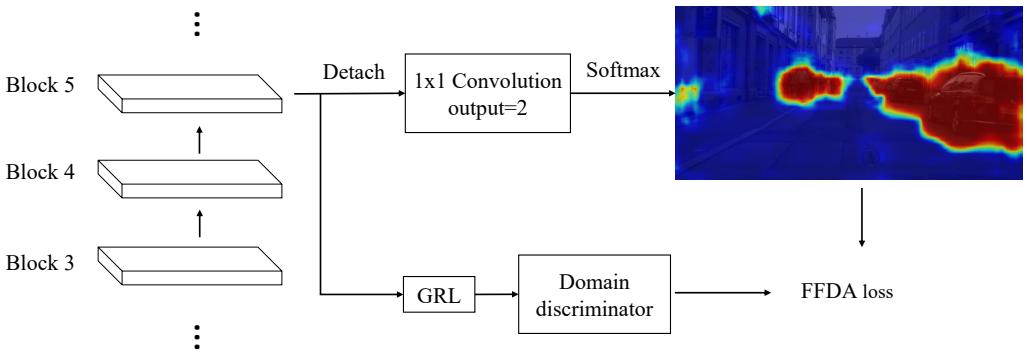

Figure 3.6: Attention-based FFDA pipeline

In this section, we propose an attention-based foreground identification as an alternative to the ROI-based solution. Innovated by works [33], [67], our attention-based FFDA generate a pixel-wise foreground mask to outline the regions for adaptation. The pipeline of this method is shown in Figure. 3.6.

The attention-based FFDA first extracts the feature from block 5, and feeds it to a one convolution layer (1x1 kernel size with output size of 2) that performs a foreground and background pixel segmentation. This layer is weakly supervised trained by using the rectangle ground-truth bounding box as the annotation. The regions inside the bounding box are set to 1 while the rest are set to 0. And we apply the standard cross-entropy loss on the output. This layer is attached to block 5 during a forward pass but detached when doing backpropagation. Such detachment is also used for training domain vectors in previous works [50], [54]. This detachment is to avoid the segmentation loss to affect the backbone. The predicted 2D map of this layer is attached with a softmax operation to translate the output to a probability mask. This mask is used to replace the mask from the ROI-based FFDA described in previous sections, and participate in the FFDA loss calculation in the image level discriminators. Unlike the ROI-based FFDA, the masks are generated in the same way for both source domain and target domain by using the proposed attention-based approach.

The experiments for this alternative foreground mask generation are shown in Table 3.6. The attention-based FFDA shows an accuracy boost of 1% mAP

Methods	C to F (mAP)	S to C (AP)
ROI-based FFDA	40.1	46.4
Attention-based FFDA	38.7	47.4

Table 3.6: Experiment results for attention-based foreground identification

on the adaptation from SIM10K to Cityscape. This result also reaches the new state-of-the-art accuracy (refer to Table 3.2). However, for the experiment of adapting from Cityscape to Foggy Cityscape, the accuracy dropped 1.4% mAP.

3.6 Compatibility tests with other pipelines

As we discussed in Section 2.1.5, pipelines of UDA for object detection can be categorized into adversarial-based, discrepancy-based, and reconstruction-based methods. We adopt a simple pipeline in discovering whether the adversarial-based FFDA could boost the accuracy further with other major pipelines in domain adaptation.

We adopt a simplified discrepancy-based pipeline that follows the works [63]. In these works, the detector is first supervised trained on source domain. Then, a set of pseudo labels is generated by making inferences with the detector on target domain. The pseudo labels are filtered and used for fine-tuning the detector on target domain. To integrate this pipeline with the FFDA, we replace the first training step on source domain with adversarial training of FFDA. We choose a simple threshold filter step for pseudo labels by placing a threshold of 0.6 for target prediction.

For integration with reconstruction-based pipeline, we follow the methods [27], [53]. We first use the Cycle-GAN to translate the images from source domain to target domain. As showed by Yu et al. [62], the original hyper parameter in Cycle-GAN shows background noise and imperfect matching. we follow the hyper parameter settings in their paper which choose a smaller window size (128x128) for translation. The adapted images are shown in Figure 3.7, top row shows the Cityscape to Foggy Cityscpae translation, bottom row shows the SIM10K to Cityscape translation. We adjust the domain discrim-

Figure 3.7: GAN translation samples

Methods	C to F (mAP)	S to C (AP)
FFDA	40.1	46.4
FFDA w/ discrepancy-based	39.2	48.0
FFDA w/ reconstruction-based	39.8	47.1

Table 3.7: Compatibility experiment result

inators used in FFDA to cooperate with this new intermediate domain that collects all translated images. We change the domain discriminator into three domain classes classification: source domain, intermediate domain, and the target domain.

The experiments of adaptation from Cityscape to Foggy Cityscape (C to F) and SIM10K to Cityscape (S to C) are shown in Table 3.7. After integrating with discrepancy-based method the accuracy of the detector improves 1.6% AP on SIM10K to Cityscape while it drops 1% mAP for Cityscape to Foggy Cityscape adaptation. The FFDA with reconstruction-based pipeline shows less improvement over the SIM10K to Cityscape experiment, but it shows less difference in accuracy compared to the FFDA on Cityscape to Foggy Cityscape. Both methods show an improvement on SIM10k to Citysc-pae dataset which shows that the FFDA is compatible with discrepancy-based and reconstruction-based pipelines and has the potential to jointly improve accuracy.

Chapter 4

UDA for WSI classification

In this chapter, we focus on the UDA approach for WSI classification. We first discuss the current works of WSI classification and the details of a previously proposed end-to-end Fisher vector coding pipeline. Next, we introduce a double adversarial adaptation approach. This proposed approach is designed for the Fisher vector coding pipeline in the effort of mitigating the domain gap in WSI classification.

4.1 Preliminary and related works

4.1.1 Standard MIL and embedding-based approach

Previous works on WSI classification typically apply Multiple-instance learning (MIL) [14]. The standard MIL approaches [7], [9], [20], [31] consider each WSI as a bag of instances. The instances inside the bag could be the sampled patches from the WSI, or tuples that combine the patches and their coordinates. A WSI is considered as a positive sample only if when some of its instances are positive. The standard MIL approaches typically perform well when the labels for the WSI are patch-based or can be indicated by the local information (e.g. colon cancer phenotype).

Another type of MIL is embedding-based approaches. The embedding-based approaches ease the assumption on the label and more suitable for WSI classification based on a global/weak label (e.g gleason score, recurrence score). Many embedding-based approaches have been proposed that are not designed and tested on WSI but general image classification [12], [41], [66]. These ap-

proaches apply various pooling strategies such as max pooling, global average pooling, etc. to aggregate the information on the extracted patches and predict the final categories for high resolution images.

Our work is established on the previous embedding-based pipeline [1]: Deep Fisher vector coding (DFVC) for WSI classification. We summarize the pipeline into four stages: First, a feature encoding stage that applies CNN to raw patches and encodes them to vectors. Second, a Fisher vector distribution encoding stage encodes the vectors with fixed vectors from the same space. Third, an average pooling stage that averages over the encoded descriptors from all the patches in a WSI. Fourth, a final classification stage utilizes a layer of linear classifier to predict the final category for the entire WSI.

4.1.2 UDA for medical imaging

To the best of our knowledge, there is no previous UDA method that applies to a WSI classification pipeline that can assign a label to the entire WSI. The closest research to ours is the work of Ren et al.[44]. They proposed a UDA solution to boost the accuracy of a DNN for predicting the gleason categories of patches extracted from the WSI. They attached a domain classifier before the final fully connected layer inside a modified AlexNet [28], and followed an adversarial training pipeline to align internal patch-wise features. They first train the DNN model to classify patches from WSIs using the source dataset. Then they adapt the features by using an adversarial loss of both domains to train the domain discriminator while allows only the adversarial loss of the target domain to update the weights of the DNN model. A regularization step is also added by feeding the DNN with a pair of target image patches every iteration. And use a one-layer classifier attached before the final fully connected layer and trained to classify the features to check If both patches are from the same WSI or not. This classifier helps to regularize the prediction so that patches from the same WSI will get a similar prediction.

Although our research shares the same motivations in mitigating the domain gap in WSI, a huge difference is that the pipeline we focus on is an end-to-end pipeline and it predicts a category for each of the WSI, while Ren

et al. [44] is only applicable to the pipelines that predict a category for each of the extracted patches. Although our research shares the same motivations in mitigating the domain gap in WSI, a huge difference is that the pipeline we focus on is an end-to-end pipeline and it predicts a category for each of the WSI, while Ren et al. [44] is only applicable to the pipelines that predict a category for each of the extracted patches.

Another difference is evaluation part. Ren et al. [44] measured the accuracy drop based on the categorical accuracy and its confusion matrix based on the extracted patches. We argue that the accuracy based on patches does not show an objective view. In their work, the label of each patch is given by the gleason category of the corresponding WSI. Many patches from a single WSI do not contain relevant information to determine the gleason categories. Therefore evaluate the adaptation based on the 'fake' patch label will put an additional bias that can further affect the result. With the support of the DFVC pipeline, we can evaluate our method based on the final prediction for each WSI instead of the biased accuracy on patches.

In addition to the only research related to WSI images, UDA has also been used for other medical imaging applications such as segmentation [61], cross-adaptation on the different types of images [64], etc. These researches typically do not hold the same characters of WSI or share a similar pipeline. Therefore, they are beyond the scope of this thesis.

4.2 Double adversarial adaptation for WSI classification

In this section, we demonstrate the proposed UDA approach that applies adversarial training to two stages of the DFVC pipeline for WSI classification.

Our solution follows an unsupervised domain adaptation approach that is trained on a labeled source WSI dataset and a target WSI dataset without label. In the source dataset, there is only a categorical label is assigned to each WSI. In this thesis, the labels are the categories of HER2 scores [42].

A full pipeline can be viewed in Figure 4.1. The consideration of having a

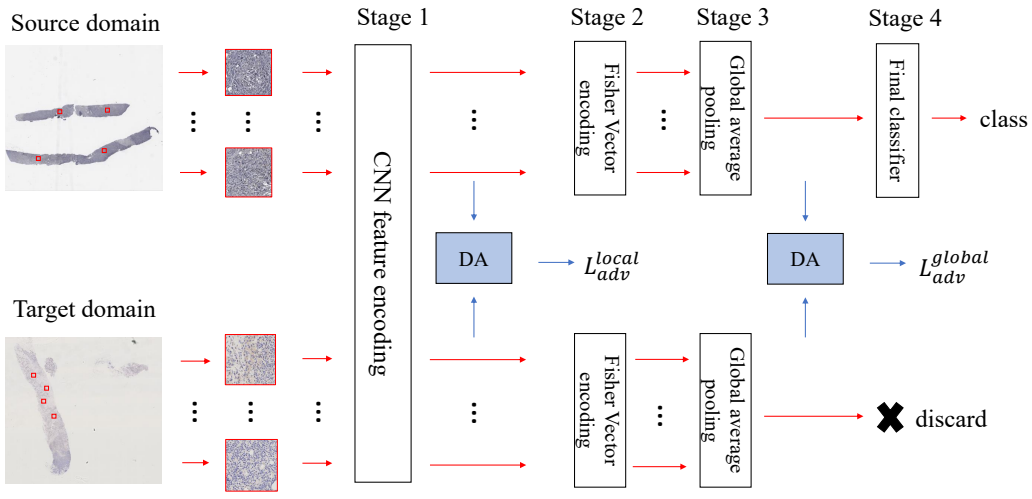

Figure 4.1: Overview of dual stages adaptation for WSI classification.

double adaptation is based on the observation that the previous pipeline has two different feature representations for the input WSI during the end-to-end process. The first one is a representation that covers a fixed-size window area randomly sampled from the WSI. This representation contains the information on cells and tissue structures inside a local region while is independent of other regions. this representation is used in both the first and second stages. In the third stage, this representation is aggregated and converted to a global representation. The global representation is a vector with significantly reduced size. It contains the abstract information that is used to generate a single prediction for the entire WSI. In training, this representation might not cover the full region of a WSI due to the limited number of samples extracted from a single WSI in each batch of data, but it can always represent the abstract information of the patches extracted from the same WSI by the global pooling in the third stage.

Based on the information differences in these stages, we propose the double stage domain adaptation that aligns the two different representations. First, the WSIs of both source domain and target domain are randomly sampled after Otsu segmentation [40] for filtering out the blank area. The sampled patches are augmented using the same procedure used by Akbarnejad et al.[1]

Input feature, local=1024x1x1, global=200x1x1
Gradient reverse layer, alpha=0.1
Convolution layer out_dim=512, kernel=1x1, stride=1, pad=None, bias=False
Relu layer
Convolution layer out_dim=2, kernel=1x1, stride=1, pad=None, bias=False
Output 2 scores

Table 4.1: Structure of global level domain discriminator

and then fed into a CNN for feature encoding.

After the feature encoding step in the first stage, we forward the features directly to a domain classifier. This domain classifier works with the patch-wise features and is responsible for adapting local distribution shift on patches from different domains. The full structure of domain adaptation part in stage one is shown in Table 4.1. The domain classifier contains two convolution layers and outputs the two domain labels - source and target. To enable adversarial learning, we attach a gradient reverse layer on the top of the domain classifier. The adversarial learning loss is a standard binary cross-entropy loss.

Besides forwarding the features to local domain adaptation part, both source and target CNN encoded features from stage one are passed to the next stages following the original training pipeline. The features are further possessed by Fisher vector encoding stage and then the global average pooling stage. The global average pooling stage aggregates the individual features so that each WSI is represented by a single vector. We insert the domain classifier to this stage to align the aggregated features. This domain classifier adapts the feature distribution shift of the entire WSI. The structure of the domain discriminator in this stage share the same configure as in the first stage with different feature size (see global input in Table 4.1).

During the training stage, the cross-entropy loss from the original pipeline and adversarial loss is combined with a balance parameter λ to update the entire model. The entire loss calculation of the proposed method can be esti-

mated as:

$$Loss = L_{CE} + \lambda(L_{adv}^{local} + L_{adv}^{global}), \tag{4.1}$$

$$L_{adv} = -D_i \log(p_i) - (1 - D_i) \log(1 - p_i) \tag{4.2}$$

L_{adv}^{local} and L_{adv}^{global} represent the adversarial loss from domain classifiers attached after the first (local) and the third (global) stage.

Chapter 5

Experiments of UDA for WSI classification

In this chapter, we show the experiment results of adapting two datasets of breast tissues WSIs. The following sections include the parameter setting details, the evaluation metrics, and the statistical results.

5.1 Implementation details

The double stage adaptation pipeline is implemented based on the original implementation of the DFVC pipeline [1]. The CNN feature encoder accepts the patches with input size of 224x224. The number of Fisher coding centers is set to 10. Balancing parameter λ is set to 0.1. We use a single Nvidia GTX 1080 Ti GPU for training and testing. A small batch size of 32 is used for both the source domain and target domain to fit in the GPU memory. Note that the global pooling step in the third stage is expected to generate the global representation, a small batch size could lead the pipeline to consider only the local information instead of the global regions, and this might result in ineffective training. To compensate for the small batch size, we set a large stepsize for gradient update. The stepsize is set to 20 and we average the accumulated loss for backpropagation when the iteration reaches the gradient update step. We choose Adam as the optimizer with a small initial learning rate of $5e^{-6}$. The training iteration is set to 50K. The evaluation metrics reported in the experiments are the top-1 accuracy and confusion matrix.

5.2 Accuracy and confusion matrices

We test our method on two HER2 breast tissue datasets. One has 500 WSIs which are collected from Alberta Cross Cancer Institution (CCI dataset). The other one has 52 WSIs which are collected from Warwick HER2 challenge [42] (Warwick dataset). The classification goal is to predict 4 categories of HER2 scores for each immunohistochemistry (IHC) stained WSI in the dataset. The categories follow the description of the Warwick challenge which are 0, 1+(negative), 2+(equivocal), and 3+(positive) given by pathologists based on the cell membrane staining pattern.

We split datasets by using half of the WSIs as training set and the other half as testing set. The dataset split keeps the ratios of the categories in training set and testing set to simulate a real scenario where there is bias among categories in datasets. In this experiment, the ratio of categories is 2:6:3:1 in CCI dataset and 1:1:1:1 in Warwick dataset.

To test the effectivness of the proposed double stage domain adaptation pipeline, we setup the experiment to adapt from a source domain dataset - CCI dataset to a target domain dataset - Warwick dataset.

The experiment in Table 5.1 includes the top-1 accuracy and confusion metrics of a model trained on source data only, a model trained with only local/first stage adaptation, a model trained with only global/third stage adaptation, a model trained with the double stage adaptation, and an oracle model trained on target data.

As we can observe from Table 5.1 (a), a significant bias is shown before using any form of adaptation, which indicates the damage of the domain shift.

Table 5.1 (c) and (d) show that both adaptation on local stage and global stage can help increase the accuracy. Compare to global stage adaptation, local stage adaptation has a better influence to all categories in the matrix. But global stage adaptation provide a better separation between the category 0 and category 3+. And from the result of Table 5.1 (e), the double stage adaptation provides the best accuracy and confusion matrix compared to the single stage adaptation in (c) and (d).

Note that training set of our CCI data is significantly large compared to the Warwick dataset, the increased accuracy in this experiment also indicates that our solution is applicable to the scenario where the model could be trained in a bigger dataset from a large institution and adapt to a smaller dataset elsewhere.

score	**0**	**1+**	**2+**	**3+**
0	0	7	0	0
1+	0	7	0	0
2+	0	7	0	0
3+	0	7	0	0

(a) Source trained, acc= 25%

score	**0**	**1+**	**2+**	**3+**
0	4	3	0	0
1+	2	3	2	0
2+	0	3	3	1
3+	0	0	1	6

(b) Oracle, acc= 51.7%

score	**0**	**1+**	**2+**	**3+**
0	7	0	0	0
1+	1	4	2	0
2+	1	2	4	0
3+	0	0	3	4

(c) Local stage adaptation, acc= 67.8%

score	**0**	**1+**	**2+**	**3+**
0	7	0	0	0
1+	6	1	0	0
2+	7	0	0	0
3+	0	1	0	6

(d) Global stage adaptation, acc=46.4%

score	**0**	**1+**	**2+**	**3+**
0	7	0	0	0
1+	3	2	2	0
2+	2	0	5	0
3+	0	0	0	7

(e) Double stages adaptation, acc= 75%

Table 5.1: Confusion matrices comparisons.

5.3 Visualization on feature alignment

We provide the t-SNE result in Figure 5.1. Refer to Section 3.2.2 for details of t-SNE. Two models are used for feature extraction and generate the t-SNE result, the source trained model and the model trained with double stage adaptation. We extract the output features from CNN encoding stage (first stage) in the DFVC pipeline. 200 features per WSI are sampled from a subset of 30 WSI of the testing set. These features are flattened and are further fed

to t-SNE for dimension reduction.

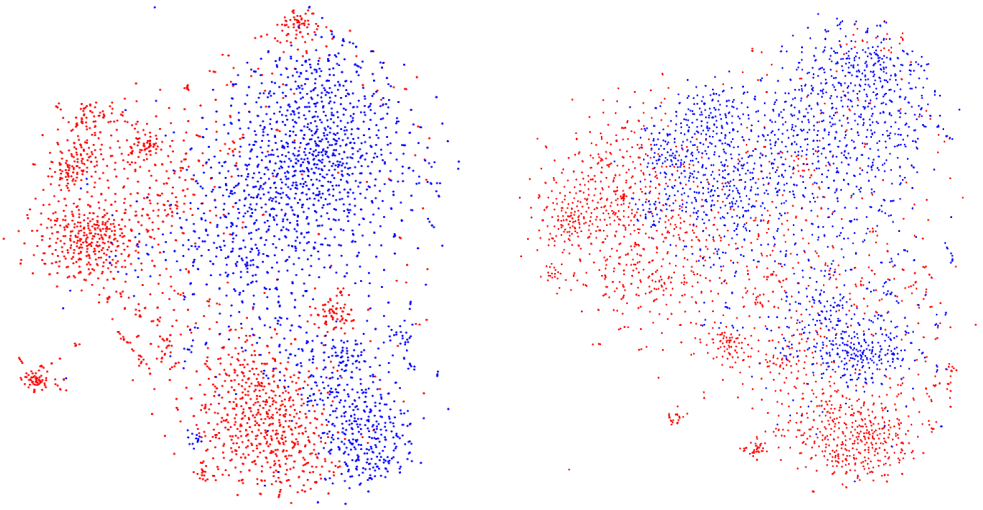

Figure 5.1: T-SNE result of testing set.

The result of source trained model without adaptation is shown on the left in Figure 5.1, and the result of the model trained with double stage adaptation is shown on the right in Figure 5.1). Unlike the result in detection (Section 3.4.3), the t-SNE result for WSI classification shows a less obvious feature alignment difference. We believe this is because the numbers of WSIs in datasets we used are small compared to the general object classification dataset. WSI dataset might show a larger distribution difference among the images. Still, we can observe that the points on the right are less aggregated around potential cluster centers than the left points. The t-SNE results along with the accuracy and confusion matrices indicate the success alignment with the adaptation.

Chapter 6

Conclusion

6.1 UDA for object detection

6.1.1 Conclusion

In this thesis, we present a novel adversarial-based approach for UDA object detection. We exploit a crucial factor - the region for adaptation on an image and the effects it could cause to the adaptation result. We notice that previous works in UDA for object detection neglected the unique task-specific characters: the sparse foreground objects and the large area of sophisticated background in an image. We hypothesis that a foreground-focused adaptation could have a significant influence on the adaptation result of object detection with our novel pipeline.

To prove the hypothesis, we propose FFDA framework that mines the loss of foreground area for feature distribution alignments in both image level and instance level. This method utilizes the labels from source domain and predictions from target domain as cues to identify the foreground area for adaptation. We collect two mining masks in each iteration during the training stage. For source domain images, we collect the groundtruth ROIs to generate the mask. And for target domain images, we collect the predicted ROIs filtered with a threshold. The masks work with the 2D score maps predicted from image level domain discriminator, and the scalar predicted from instance level domain discriminator. By doing this procedure, the discriminators allow only the identified foreground region to participate in adversarial loss calculation.

Finally, a multi-adversarial structure is applied to further boost the feature alignment.

Experiments on several datasets for autonomous driving applications show the superior performance of the FFDA method compared to previous methods. The ablation study visually validates the alignment by using t-SNE and Grad-CAM. And the result shows the foreground-focused adaptation can provide a better alignment on features and more decisive information for the detector. A detailed experiment of comparing performance of each component in full feature alignment vs. foreground-focused alignment settings is also provided. The result shows that adapting the background area in any components will cause degeneration in the accuracy, and consistently impose the foreground-focused adaptation can achieve the best accuracy.

6.1.2 Future works

An alternative foreground identification and two mixed-pipeline approaches are briefly discussed in Section 3.5 & 3.6. For the alternative foreground identification, we only adopt a simple self-attention mechanism. The accuracy of the proposed attention-based FFDA pipeline is increased on one experiment while decreased on another. In future work, we could expand this self-attention for mask generation pipeline to further identify the 'perfect' regions for adaptation. The same observation can also be found in the experiments of FFDA with the discrepancy-based pipeline, and FFDA with the reconstruction-based pipeline. The mixed-pipeline approach is also an interesting topic and has the potential to reach higher accuracy.

Another future work is to increase the flexibility of the FFDA. Most of the existing works include the proposed FFDA choose the two-stage detector -Faster RCNN as the base detector for UDA research. Recently developed detectors, such as Carion et al.[6] which use transformer for object detection, could make it hard for current domain adaptation methods to be deployed to their pipelines. It has become important to cover various types of detectors in future work to show the flexibility of UDA solutions towards all kinds of detectors.

6.2 UDA for WSI classification

6.2.1 Conclusion

This thesis focuses on the domain shift problem that exists in WSI classification task. Like many other tasks, WSI classification also suffers the accuracy drop from the domain gap. However, there is a lack of end-to-end pipelines for WSI classification, and thus the studies in mitigating the domain gap issue can only work with a patch-wise classification pipeline.

Recently, an end-to-end deep fisher coding pipeline is proposed and provides an opportunity to consider adaptation in both local to global aspects. Built on this pipeline, we propose a double stage alignment that utilizes the domain classifier to align the inner features extracted from this pipeline.

We consider both the local distribution shift between the patch-wise representations of source domain and target domain, as well as the global distribution shift between the global representations. This proposed double stages adversarial adaptation inserts domain classifiers to the CNN encoding stage and the fisher vector coding stage in the previous pipeline.

The predicted confusion matrices in the experiment show that the bias between the WSI datasets is reduced by conduct adversarial learning. And a double alignment in different stages achieves a better result than a single alignment in one stage.

6.2.2 Future works

The improvement of the UDA approach for WSI classification requires joint effort from the classification part and the domain adaptation part.

For the classification part, the attention-based mechanism may boost the accuracy of the previous end-to-end pipeline. It may be possible to outlines the decisive regions by observing the gradient changed inside the end-to-end pipeline. The character of scattered diagnostic info of the WSI is similar to the sparse foreground setting in object detection. Therefore, it is possible to apply the same foreground-focused adaptation scheme to the UDA for WSI classification if we can separate the diagnostic regions in the WSI.

For the adaptation part, One of the future works can be using a different loss function. Unlike the general object classification task, the tissue structures from the same organ under the digital scan are similar. Loss function with regularization on structure differences might be more suitable for the WSI adaptation.

Another place for improvement is to collect more datasets that are suitable for adaptation experiments. The WSI dataset must share the same annotation categories and have enough WSIs to train the end-to-end pipeline. It is necessary to collect more datasets that covers varies scenarios such as different institution, different scanning machine, etc.

References

[1] A. Akbarnejad, N. Ray, and G. Bigras, "Deep fisher vector coding for whole slide image classification," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 2021.

[2] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, IEEE, 2007, pp. 77–82.

[3] V. F. Arruda, T. M. Paixão, R. F. Berriel, A. F. De Souza, C. Badue, N. Sebe, and T. Oliveira-Santos, "Cross-domain car detection using unsupervised image-to-image translation: From day to night," in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8.

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

[5] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2724–2732.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*, Springer, 2020, pp. 213–229.

[7] H. Chen, X. Han, X. Fan, X. Lou, H. Liu, J. Huang, and J. Yao, "Rectified cross-entropy and upper transition loss for weakly supervised whole slide image classifier," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 351–359.

[8] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[9] M. Combalia and V. Vilaplana, "Monte-carlo sampling applied to multiple instance learning for histological image classification," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 2018, pp. 274–281.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[11] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[12] T. Durand, N. Thome, and M. Cord, "Weldon: Weakly supervised learning of deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4743–4752.

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[14] J. R. Foulds and E. Frank, "A review of multi-instance learning assumptions," 2010.

[15] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *International Conference on Machine Learning*, 2015.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[17] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Pacific Rim international conference on artificial intelligence*, Springer, 2014, pp. 898–904.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

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

[20] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, "Patch-based convolutional neural network for whole slide tissue image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2424–2433.

[21] H.-K. Hsu, W.-C. Hung, H.-Y. Tseng, C.-H. Yao, Y.-H. Tsai, M. Singh, M.-H. Yang, W. Treible, P. Saponaro, Y. Liu, *et al.*, "Progressive domain adaptation for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1–5.

[22] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5001–5009.

[23] J. Jiang, Y.-C. Hu, N. Tyagi, P. Zhang, A. Rimner, G. S. Mageras, J. O. Deasy, and H. Veeraraghavan, "Tumor-aware, adversarial domain adaptation from ct to mri for lung cancer segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 777–785.

[24] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" *arXiv preprint arXiv:1610.01983*, 2016.

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

[26] S. Kim, J. Choi, T. Kim, and C. Kim, "Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6092–6101.

[27] T. Kim, M. Jeong, S. Kim, S. Choi, and C. Kim, "Diversify and match: A domain adaptive representation learning paradigm for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 456–12 465.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[29] M. W. Lafarge, J. P. Pluim, K. A. Eppenhof, and M. Veta, "Learning domain-invariant representations of histological images," *Frontiers in medicine*, vol. 6, p. 162, 2019.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[31] Z. Li, R. Tao, Q. Wu, and B. Li, "Da-refinenet: A dual input whole slide image segmentation algorithm based on attention," *arXiv preprint arXiv:1907.06358*, 2019.

[32] C.-T. Lin, "Cross domain adaptation for on-road object detection using multimodal structure-consistent image-to-image translation," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 3029–3030.

[33] C. Lin, J. Lu, G. Wang, and J. Zhou, "Graininess-aware deep feature learning for pedestrian detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 732–747.

[34] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[38] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[39] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," *arXiv preprint arXiv:1711.02536*, 2017.

[40] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[41] N. Passalis and A. Tefas, "Learning bag-of-features pooling for deep convolutional neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5755–5763.

[42] T. Qaiser, A. Mukherjee, C. Reddy Pb, S. D. Munugoti, V. Tallam, T. Pitkäaho, T. Lehtimäki, T. Naughton, M. Berseth, A. Pedraza, *et al.*, "Her 2 challenge contest: A detailed assessment of automated her 2 scoring algorithms in whole slide images of breast cancer tissues," *Histopathology*, vol. 72, no. 2, pp. 227–238, 2018.

[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[44] J. Ren, I. Hacihaliloglu, E. A. Singer, D. J. Foran, and X. Qi, "Adversarial domain adaptation for classification of prostate histopathology whole-slide images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 201–209.

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[46] A. L. Rodriguez and K. Mikolajczyk, "Domain adaptation for object detection via style consistency," *British Machine Vision Conference*, 2019.

[47] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," *arXiv preprint arXiv:2005.03778*, 2020.

[48] A. Rosenfeld, R. Zemel, and J. K. Tsotsos, "The elephant in the room," *arXiv preprint arXiv:1808.03305*, 2018.

[49] A. RoyChowdhury, P. Chakrabarty, A. Singh, S. Jin, H. Jiang, L. Cao, and E. Learned-Miller, "Automatic adaptation of object detectors to new domains using self-training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 780–790.

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

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

[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[53] Y. Shan, W. F. Lu, and C. M. Chew, "Pixel and feature level based domain adaptation for object detection in autonomous driving," *Neurocomputing*, vol. 367, pp. 31–38, 2019.

[54] Z. Shen, H. Maheshwari, W. Yao, and M. Savvides, "Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses," *arXiv preprint arXiv:1911.02559*, 2019.

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

[56] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*, Springer, 2016, pp. 443–450.

[57] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J.-M. Bokhorst, F. Ciompi, and J. van der Laak, "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Medical image analysis*, vol. 58, p. 101 544, 2019.

[58] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[59] R. Xie, F. Yu, J. Wang, Y. Wang, and L. Zhang, "Multi-level domain adaptive learning for cross-domain detection," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[60] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3964–3973.

[61] J. Yang, N. C. Dvornek, F. Zhang, J. Chapiro, M. Lin, and J. S. Duncan, "Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 255–263.

[62] F. Yu, D. Wang, Y. Chen, N. Karianakis, P. Yu, D. Lymberopoulos, and X. Chen, "Unsupervised domain adaptation for object detection via cross-domain semi-supervised learning," *arXiv preprint arXiv:1911.07158*, 2019.

[63] Y. Yu, X. Xu, X. Hu, and P.-A. Heng, "Dalocnet: Improving localization accuracy for domain adaptive object detection," *IEEE Access*, vol. 7, pp. 63 155–63 163, 2019.

[64] Y. Zhang, H. Chen, Y. Wei, P. Zhao, J. Cao, X. Fan, X. Lou, H. Liu, J. Hou, X. Han, *et al.*, "From whole slide imaging to microscopy: Deep microscopy adaptation network for histopathology cancer image classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 360–368.

[65] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9259–9266.

[66] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[67] C. Zhou, M. Wu, and S.-K. Lam, "Ssa-cnn: Semantic self-attention cnn for pedestrian detection," *arXiv preprint arXiv:1902.09080*, 2019.

[68] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

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

[70] C. Zhuang, X. Han, W. Huang, and M. R. Scott, "Ifan: Image-instance full alignment networks for adaptive object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Appendix A

More qualitative result

Here we provide more qualitative results on Cityscape to Foggy Cityscape (Figure A.1), SIM10K to Cityscape (Figure A.2) and KITTI to Cityscape (Figure A.3) datasets. From left to right: results from a Faster RCNN trained in source domain only, MLDA, and ours. Zoom in for details. FFDA is capable of detecting more objects in target domain compared to a full feature alignment from MLDA, and a source trained only detector.

Figure A.1: Cityscape to Foggy Cityscape adaptation

Figure A.2: SIM10K to Cityscape adaptation

Figure A.3: KITTI to Cityscape adaptation