2023

# Multi-Domain Adaptation for Image Classification, Depth Estimation, and Semantic Segmentation

Yu Zhang
*University of Kentucky*, yuzh03@gmail.com

Digital Object Identifier: https://doi.org/10.13023/etd.2023.038

Yu Zhang, Student

Dr. Brent Harrison, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

# Multi-Domain Adaptation for Image Classification, Depth Estimation, and Semantic Segmentation

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for the
degree of Doctor of Philosophy in the
College of Engineering at the
University of Kentucky

By
Yu Zhang
Lexington, Kentucky

Director: Dr. Brent Harrison
Assistant Professor of Computer Science

Co-Director: Dr. Nathan Jacobs
Professor of Computer Science

Lexington, Kentucky 2023

# ABSTRACT OF DISSERTATION

## Multi-Domain Adaptation for Image Classification, Depth Estimation, and Semantic Segmentation

The appearance of scenes may change for many reasons, including the viewpoint, the time of day, the weather, and the seasons. Traditionally, deep neural networks are trained and evaluated using images from the same scene and domain to avoid the domain gap. Recent advances in domain adaptation have led to a new type of method that bridges such domain gaps and learns from multiple domains.

This dissertation proposes methods for multi-domain adaptation for various computer vision tasks, including image classification, depth estimation, and semantic segmentation. The first work focuses on semi-supervised domain adaptation. I address this semi-supervised setting and propose to use dynamic feature alignment to address both inter- and intra-domain discrepancy. The second work addresses the task of monocular depth estimation in the multi-domain setting. I propose to address this task with a unified approach that includes adversarial knowledge distillation and uncertainty-guided self-supervised reconstruction. The third work considers the problem of semantic segmentation for aerial imagery with diverse environments and viewing geometries. I present CrossSeg: a novel framework that learns a semantic segmentation network that can generalize well in a cross-scene setting with only a few labeled samples. I believe this line of work can be applicable to many domain adaptation scenarios and aerial applications.

KEYWORDS: domain adaptation, classification, depth estimation, semantic segmentation

Author's signature: Yu Zhang

Date: March 9, 2023

# Multi-Domain Adaptation for Image Classification, Depth Estimation, and Semantic Segmentation

By
Yu Zhang

Director of Dissertation: Dr. Brent Harrison

Co-Director of Dissertation: Dr. Nathan Jacobs

Director of Graduate Studies: Dr. Simone Silvestri

Date: March 9, 2023

# ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Nathan Jacobs. Dr. Jacobs was always there to listen and give advice whenever I encountered obstacles, even beyond research. He not only helped me seek funding opportunities from multiple sources to support my research but was a patient and strong mentor and inspired me to work hard all the time. I would have never accomplished what I have without his support and guidance.

I sincerely thank Dr. Brent Harrison, for serving as my committee chair, and the rest of my committee members, Dr. Ramakanth Kavuluru, Dr. Qiang Ye, and Dr. Mihai Tohaneanu for their valuable feedback and discussions, which helped me complete this dissertation and present my work in a better way. I would also like to thank Dr. Xiaoqin Wang, who supported my medical imaging research, and Dr. Yuanyuan Su, who guided me to explore the interesting field of deep learning in astronomy. I am grateful to the DGS Dr. Simone Silvestri and all the other faculties in the computer science department, for their advice and support in different stages during my Ph.D. studies.

My grateful thanks are due to my lab peers, including Gongbo Liang, Usman Rafique, Xin Xing, Zach Bessinger, Tawfiq Salem, Connor Greenwell, Hunter Blanton, Benjamin Brodie, Armin Hadzic, and the rest of the lab members. They provided helpful feedback and discussions which helped me in different ways. I would like to express my gratitude to my parents and my grandfather, who almost supported every decision I made in my life, including studying abroad, and my girlfriend Yeqi, who always supported and encouraged me no matter how hard the situation was. Without them, I would have never become who I am today.

To my grandfather.

## TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# Introduction

Deep learning has achieved great successes in computer vision, especially in image classification [69], object detection [47], and semantic segmentation [80]. The key factor of deep learning is a large amount of labeled training data. Annotating source training data, however, is time-consuming and expensive. Moreover, massive labeled source data cannot always guarantee conventional neural networks to generalize well because it cannot always represent the target environment, especially when there is a potential mismatch or bias in the training data. For example, a deep learning system of self-driving cars trained on sunny environment data may not perform well in the rainy/snowy/foggy environment because of the domain shift [155]. In this work, I develop frameworks of multiple domain adaptation scenarios for various computer vision tasks, including image classification, depth estimation, and semantic segmentation.

## 1.1 Understanding Domain Shift

Since the prevalence of machine learning, the most dominant type of machine learning method is supervised learning no doubt. Fully supervised learning methods achieved much success across different applications in many areas. However, it assumes that the training and test data are selected from the same distribution. However, this assumption is often invalid for many real-world scenarios, and the model trained only on the source domain will likely experience a performance drop when tested on data from other target domains. This performance drop caused by the mismatch between train and test data is usually called a domain gap.

To bridge this domain gap and alleviate the dataset mismatch and bias mentioned above, I consider *Domain Adaptation* approaches. *Unsupervised Domain Adaptation* utilizes labeled training data and unlabeled target data to build a neural network that performs well on the target environment [122, 38]. In most of the real-world scenarios, however, there exists a small number of labeled samples in the target domain for us to use [22, 136], so how to make the most advantage of those data becomes another vital problem, which is addressed by *Semi-Supervised Domain Adaptation*. Even though multiple domain adaptation methods have been proposed recently [53, 111, 98, 64], the domain shift and mismatch is still a challenging, open problem, especially on multimodal data including both 2D images and 3D point cloud data. Many efforts have been devoted to improving network generalization ability and bridging the domain gap [7, 18, 43]. Among them, adversarial training [116, 145], which trains networks with source samples and target/adversarial samples alternatively, stands as one of the most effective methods. In this work, I propose multiple variants of adversarial learning methods to tackle both unsupervised and semi-supervised domain adaptation problems for different vision tasks.

Another factor that affects the generalization ability of deep neural networks is the multi-modality of training data. For example, in 2D vision, pre-training a network on a rich source dataset (e.g., ImageNet [21]) can boost the performance once fine-tuned on a much smaller target dataset. Recently, learning-based 3D vision methods have made much progress [1, 17, 20]. However, training from scratch on the target dataset is still the dominant strategy [28, 36] compared to its 2D counterpart, which takes great advantage of pre-training. How to train the networks appropriately using multimodal data is a crucial point for many applications. In this work, I not only discuss 2D vision tasks including image classification and semantic segmentation but also explore the 3D task depth estimation. My goal is to provide a general multi-domain adaptation solution for various vision tasks.

## 1.2 Classification, Semantic Segmentation, and Depth Estimation

In this section, I introduce three fundamental computer vision tasks, image classification, semantic segmentation, and depth estimation. I especially consider their applications in aerial and remote sensing data.

Image classification is the most fundamental task among the three mentioned above that attempts to understand an entire image as a whole. The purpose is to classify the input image by labeling it with one specific category. Typically, image classification can be applied to images in which only one predominant object appears. In contrast, semantic segmentation considers pixel-level classification and usually analyzes more realistic scenarios in which multiple objects may exist in the image. Depth estimation shares the same idea with semantic segmentation by predicting at the pixel level, but instead of classifying pixels into categories, depth estimation regresses the distance between the pixel in the scene and the camera.

Traditionally, hand-crafted features are widely used in these tasks before the prevalence of deep neural networks. Those hand-crafted features are mostly designed on the basis of domain-specific knowledge, and the problem is that unfeasible to address the need of considering all of the details embedded in all forms of real data via the use of pre-designed hand-crafted features. Instead of relying on shallow manually engineered features, deep neural networks are able to automatically learn informative representations of raw input data with multiple levels of abstraction.

Instead of considering the predominant object in the image and assigning it to a specific label, semantic segmentation explores pixel-wise classification and considers multiple objects in the input data. Semantic segmentation is one of the most challenging tasks in automatic visual understanding, leading to a deeper understanding of the image content if compared with simpler problems like image classification or object detection. Historically, semantic segmentation has moved its origins as an enriched representation and understanding of the scene with respect to the simpler task of image classification: the advent of novel problems to address requiring a higher level of interpretation of the scenes and the possibility to accomplish it, thanks to novel architectures and paradigms (e.g., deep learning), have paved the way to the wide success of semantic image segmentation. While image classification allows classifying what is contained in an image at a global level (i.e., one label is assigned to each image), semantic image segmentation generates a pixel-wise mask of each object in the images (i.e., one label is assigned to each pixel of each image). Being the former a much simpler task, it has been tackled for a long time with both traditional techniques based on a feature extraction step (e.g., using SIFT or other feature extractors) followed by a classification stage (e.g., using SVM, LDA, or Random Forests) and, more recently, with deep learning ones. For this reason, some early-stage works in semantic segmentation build up from classification works, adapting and extending them. The most recent state-of-the-art approaches rely on an autoencoder structure, composed of an encoder and a decoder in order to extract global semantic clues while retaining input spatial dimensionality.

Starting from the well-known Fully Convolutional Networks (FCN) architecture [80], many models have been proposed, such as PSPNet [147], DRN [139] and the various versions of the DeepLab architecture [10, 13, 12]. These models can achieve impressive performance, but this is strictly related to the availability of a massive amount of labeled data required for their training. For this reason, even though the pixel-wise annotation procedure is highly expensive and time-consuming, many datasets have been created: for example, Cityscapes [19] and Mapillary [92] for urban scenes; Pascal VOC [31], MS-COCO [76] and ADE20K [150] for visual objects in common contexts. In light of these considerations, many recent works try to exploit knowledge extracted from other sources or domains, where labels are plentiful and easily accessible, to reduce the amount of required manually annotated data. My work focuses on using a limited number of labeled data to train a model that can generalize well to other new domains.

Depth estimation is another important task for many vision scenarios including autonomous vehicles, UAVs, robotics, and remote sensing. Recent research on monocular depth estimation can be categorized into three groups [63]: supervised, weakly supervised, and self-supervised. Supervised depth estimation networks [26, 27, 34] require a larger volume of ground-truth depth annotations. These methods formulate depth estimation as a regression problem and directly learn from the supervised losses. The weakly-supervised line of depth estimation works [82, 70] do not require depth annotations but require other labels, including semantic labels or odometry. For instance, DESC [82] proposes an unsupervised domain adaptation depth estimation network that uses ground-truth semantic labels from both source and target domains to enforce the consistency between the predictions from a semantic branch and a depth estimation branch. CoMoDA [70] adds a velocity loss to Monodepth2 [49] and performs inference-time adaption to unseen test data. Pseudo-labeling-based methods [154, 15, 134] generate pseudo labels from internet photo collections by using the ground-truth ordinal depth information as a cue and leveraging multi-view stereo reconstruction algorithms. The self-supervised group explores learning algorithms using either rectified stereo image pairs [40, 48] or monocular video sequences [151, 137, 49, 63] as training data. The video-based depth estimation methods [77, 84] use consecutive monocular frames to estimate depth during inference, making the assumption that scenes are mostly rigid. There are also several works on monocular depth completion [29, 87, 135] that have been proposed to capitalize on sparse depth maps with corresponding images, resulting in dense depth estimations. My proposed method uses monocular video sequences as training data and can be considered as a combination of supervised and self-supervised approaches.

Monocular depth estimation methods typically consider a single domain, usually ground-level indoor and outdoor images, without considering how they generalize to other domains, such as aerial images. A recent work [88] directly applies a variant of Monodepth2 [49] to UAV videos and achieves reasonable results. However, this work is also limited to a single domain of aerial images, and it does not consider adapting the model to both ground-level and aerial images simultaneously.

## 1.3 Contributions

The focus of this research is learning from limited labeled data from multiple domains for image classification, semantic segmentation, and depth estimation. I develop three novel frameworks to address these major issues and solve them. I also conduct a variety of experiments to show the effectiveness of the proposed methods. The contributions of this dissertation are as follows:

- **Semi-Supervised Domain Adaptation**: I address this semi-supervised setting and propose to use dynamic feature alignment to address both inter- and intra-domain discrepancy. I propose to align the target features to a set of dynamically updated class prototypes, which I use both for minimizing divergence and pseudo-labeling. By updating based on class prototypes, I avoid problems that arise in previous approaches due to class imbalances.

- **Multi-Domain Depth Estimation**: I address the task of monocular depth estimation in the multi-domain setting. Given a large source dataset with ground-truth depth maps, and a set of unlabeled target datasets, my goal is to create a model that works well on unlabeled target datasets across different scenes. This is a challenging problem when there is a significant domain shift, often resulting in poor performance on the target datasets. I propose to address this task with a unified approach that includes adversarial knowledge distillation and uncertainty-guided self-supervised reconstruction. My approach significantly improves upon conventional domain adaptation baselines and does not require additional memory as the number of target sets increases.

- **Few-Shot Semantic Segmentation**: I consider the problem of semantic segmentation for aerial imagery with diverse environments and viewing geometries. Conventional semantic segmentation approaches can only recognize the classes at test time that have appeared in the training set and are hard to generalize well to unseen object categories. This is a significant limitation for autonomous systems, especially for those deployed in a realistic real-time setting, e.g., unmanned aerial vehicles (UAVs). In this work, I address the task of few-shot semantic segmentation for different aerial scenes. I present CrossSeg: a novel framework that learns a semantic segmentation network that can generalize well in a cross-scene setting with only a few labeled samples. Instead of using a set of fixed prototypes, CrossSeg offers high-quality probabilistic prototypes which can not only represent different semantic classes but can also enhance the huge variations in aerial images. Experiments show that my method performs better than previous baselines without requiring extensive tuning.

The contribution of this dissertation is an improved framework for domain adaptation that is more robust than previous approaches while also allowing part of the network to be reused by other applications without massive modifications.

### 1.4 Dissertation Outline

The remainder of this document consists of the following chapters:

- **Chapter 2** provides a technical background that is necessary for understanding the work in this dissertation. I provide an overview of related background knowledge and research in the convolutional neural net, semi-supervised learning, self-supervised reconstruction, and prototypical learning.

- **Chapter 3** introduces a semi-supervised domain adaptation framework, which uses dynamic feature alignment to address both inter- and intra-domain discrepancy. The key contribution is to align the target features to a set of dynamically updated class prototypes, which I use both for minimizing divergence and pseudo-labeling. By updating based on class prototypes, I avoid problems that arise in previous approaches due to class imbalances.

- **Chapter 4** proposes the task of monocular depth estimation in the multi-domain setting. Given a large source dataset with ground-truth depth maps, and a set of unlabeled target datasets, my goal is to create a model that works well on unlabeled target datasets across different scenes. This is a challenging problem when there is a significant domain shift, often resulting in poor performance on the target datasets. I propose to address this task with a unified approach that includes adversarial knowledge distillation and uncertainty-guided self-supervised reconstruction.

- **Chapter 5** considers the problem of semantic segmentation for aerial imagery with diverse environments and viewing geometries. I introduce a novel few-shot learning-based method for the semantic segmentation of aerial images. My method can perform segmentation for unseen object categories with only a few annotated samples. My method proposes to model prototypes in a probabilistic way instead of using fixed prototypes for each class.

- **Chapter 6** summarizes the contributions of this dissertation. I highlight the key findings as well as discuss the significance of each contribution. Finally, I discuss several possible future research directions.

# Chapter 2

# Technical Background

In this chapter, to help readers understand the proposed study, relevant technical background information is provided. The concept of convolutional neural networks, a type of neural network commonly used for image-related tasks, is first introduced, followed by the description of semi-supervised learning. Semi-Supervised learning is widely used in deep learning model training, especially when the number of labeled data is limited. Then the method of self-supervised reconstruction is discussed, which is a novel modeling tool used for video sequences. Finally, this chapter ends with prototypical learning, a framework that learns representative feature vectors from each class and computes the distances between the input data and the prototypes. Prototypical learning has been widely used in domain adaptation and few-shot learning settings.

## 2.1 Convolutional Neural Networks

Biological neural networks in human brains inspired the invention of a computing model called an artificial neural network. Convolutional Neural Network (CNN) is a type of neural network, first introduced in the 1980s by Neocognitron [35]. This architecture proposed the two basic types of layers in CNNs: convolutional layers and downsampling layers. In the 1990s, LeNet [71], a gradient-based pioneering CNN, was proposed. Since Alex Krizhevsky designed the AlexNet structure in 2012 [69], CNNs have dominated the field of computer vision, achieving significant success. AlexNet outperforms the previous state-of-the-art methods significantly, dropping the classification error from $26\%$ to $15\%$ on the task of classifying 1.2 million images to 1,000 classes. Since then, CNNs have generated much excitement in research and industry. Researchers in computer vision continue using CNNs and have shown great success on traditional computer vision problems: image classification [69, 83], object detection [47, 107], and semantic segmentation [80, 96]. Furthermore, different big companies started employing deep learning in their services, such as face recognition at Facebook, photo search at Google, and product recommenda-

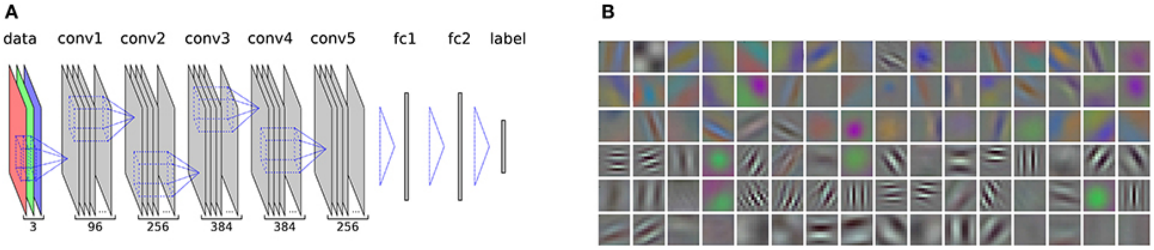


Figure 2.1: Convoluational neural network (CNN) structure (A), and feature visualizations (B) [6].

tion systems at Amazon. One of the main reasons behind the great performance of CNNs is their ability to learn hierarchical feature representation of the input images while traditional methods use hand-engineered features.

How can CNNs learn the hierarchical features representation of an image? Computers see an image as an array of numbers with size equals $width \times height \times channels$. For example, in image classification, the input is a matrix, and the expected output is a probability distribution over the different classes. To perform image classification, CNNs look for low-level features such as edges and curves in the training images and then construct abstract concepts through a series of linear and nonlinear operations achieved by a combination of layers.

The major components of a CNN model are several convolutional and subsampling layers optionally followed by fully connected layers. The excellent performance of CNNs most of the time comes from problems that involve learning discriminative models that usually map high-dimensional data (e.g., image) to a class label as shown in Figure 2.1 (A), and CNNs achieve this by extracting useful features showed in Figure 2.1 (B). This learning approach, supervised learning, for training convolutional neural networks depends on large amounts of labeled samples as training data. Conventional CNNs including AlexNet [69] and ResNet [55] only works for 2D images. For 3D data, for instance, point cloud and RGBD data, several widely used architectures are designed to handle these unordered point sets in 3D space. In this dissertation, we propose different variants of CNNs to solve image classification, depth estimation, and semantic segmentation problems.

## 2.2 Domain Adaptation

Deep neural networks have achieved impressive performance on a wide range of tasks, including image classification, semantic segmentation, and object detection. However, models often generalize poorly to new domains, such as when a model trained on indoor imagery is used to interpret an outdoor image. *Domain Adaptation* (DA) methods aim to take a model trained on a label-rich source domain and make it generalize well to a label-scarce target domain. Recently, most studies on domain adaptation have focused on the *Unsupervised Domain Adaptation* (UDA) setting, in which no labeled target data is available.

### 2.2.1 A Clinical Case for Unsupervised Domain Adaptation

One of our recent works provides a simple yet effective UDA solution to medical imaging [144].

**Overview** Generalization is one of the key challenges in the clinical validation and application of deep learning models to medical images. Studies have shown that such models trained on publicly available datasets often do not work well on real-world clinical data due to the differences in patient population and image device configurations. Also, manually annotating clinical images is expensive. In this work, we propose an unsupervised domain adaptation (UDA) method using Cycle-GAN to improve the model's generalization ability without using any additional manual annotations.

We know if we train a deep learning model on a labeled dataset A (source domain), it may achieve high performance on A but low performance on an unlabeled dataset B (target domain) because A and B may have different attributes. We hypothesize the UDA method will improve the model's performance on B while maintaining the high performance on A.

**Dataset** The public mammogram dataset Digital Database of Screening Mammography (DDSM) [56] and a private mammogram dataset, UKY [131], are used in this work. These two datasets have different attributes: DDSM contains digitalized screen film mammograms and UKY contains full-field digital mammograms recently collected from a comprehensive breast imaging center. Several recent works explored the different attributes of those two or other similar datasets [146, 74, 142]. In this work, we use 1860 positive and 2781 negative images from DDSM and 1922 positive and 2330 negative images from UKY. We split the data in 80% for training and 20% for testing.

**Method** Figure 2.2 illustrates our UDA method. We first train the Cycle-GAN [152] on unpaired images without any labels, then we synthesize DDSM data from UKY data to generate training samples in the target domain. Finally, we train a deep neural network on a mixture of UKY and synthesized DDSM images. We compared our UDA method with the baseline method, which trains on one dataset and directly tests on another dataset. In addition, we train the models on labeled DDSM and synthesized UKY by switching the source and target domains for a two-way verification.

**Experimental Results** Our results are summarized in Table 2.1. Two off-the-shelf architectures are used for evaluation: AlexNet [69] and ResNet [55]. When training and testing

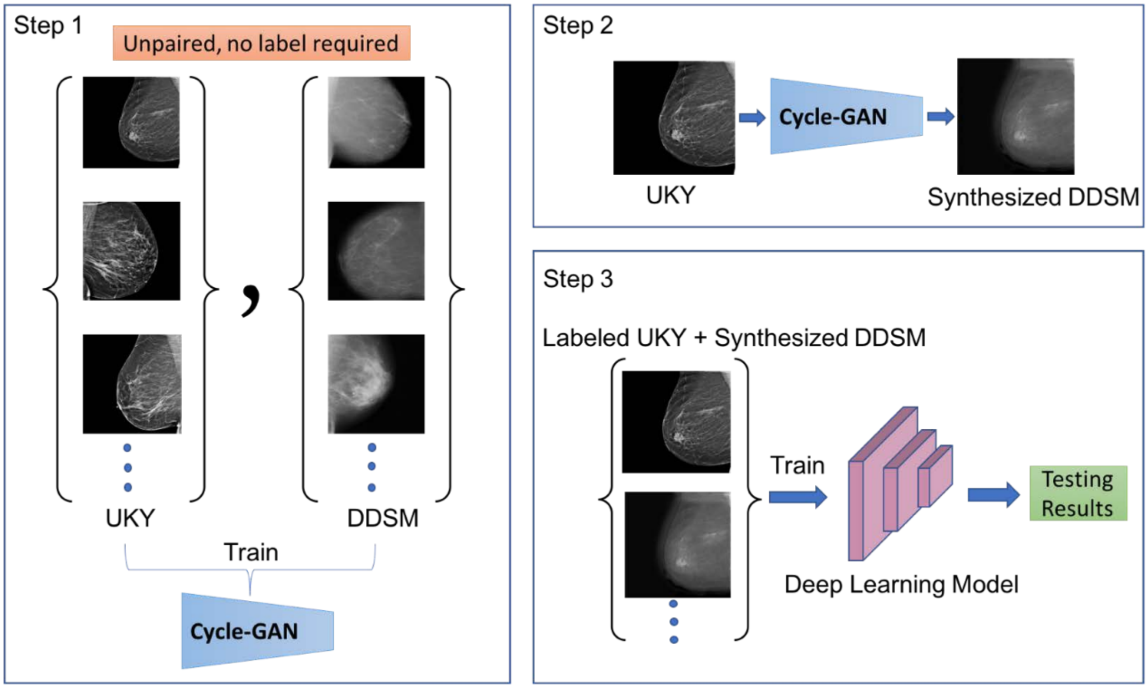


Figure 2.2: Stepwise illustration of our unsupervised domain adaptation(UDA) method. Step 1) train Cycle-GAN by using unpaired, unlabeled UKY and DDSM datasets; Step 2) translate UKY to DDSM; 3) train deep learning models by using UKY and synthesized DDSM.

Table 2.1: Testing Results of Different Methods.

| Training Set | Testing Set | Mean auROC (95% Confidence Interval) | | | |
|---|---|---|---|---|---|
| | | AlexNet | | ResNet50 | |
| | | Baseline | UDA | Baseline | UDA |
| UKY | DDSM | 0.516 ± 0.004 | **0.601** ± 0.005 | 0.624 ± 0.004 | **0.672** ± 0.002 |
| | UKY | 0.785 ± 0.003 | 0.769 ± 0.007 | 0.836 ± 0.008 | 0.869 ± 0.016 |
| DDSM | UKY | 0.491 ± 0.007 | **0.578** ± 0.002 | 0.565 ± 0.002 | **0.674** ± 0.003 |
| | DDSM | 0.673 ± 0.015 | 0.653 ± 0.024 | 0.762 ± 0.008 | 0.759 ± 0.012 |

on different datasets, UDA achieves significant improvement compared to the baseline. For instance, when we trained on UKY and tested on DDSM with AlexNet, the baseline only achieved 0.516 auROC while UDA achieved 0.601 auROC. The table also shows when training and testing on the same dataset, UDA maintains similarly high performance, which verified our hypothesis.

**Conclusion** Our results show that the proposed UDA method improves deep learning models' generalization without requiring expensive manual annotations. However, there is still room for improvement. We expect combining improved versions of Cycle-GAN with small amounts of labeled data in the target domain will help bridge the gap.

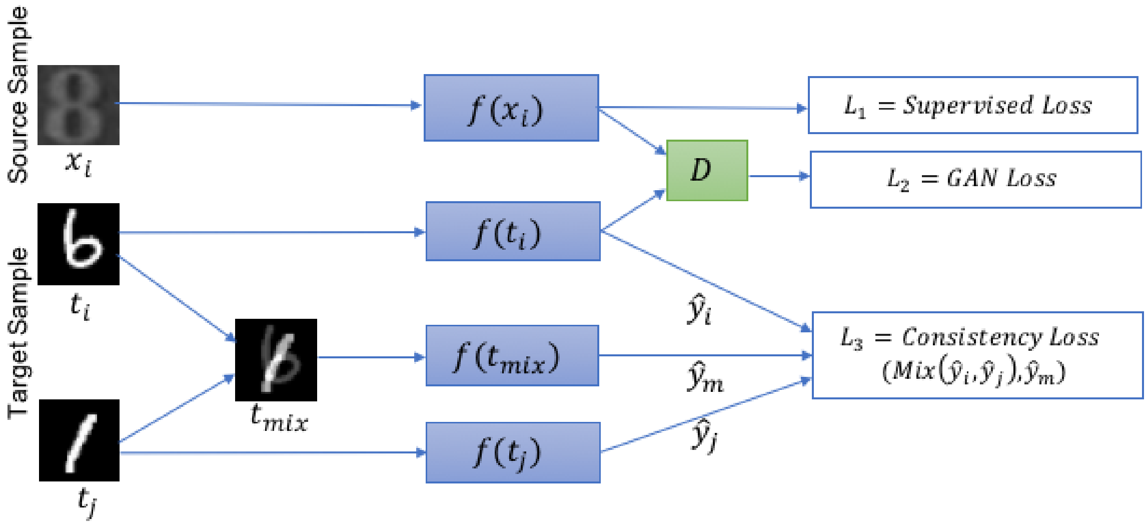


Figure 2.3: SSDA via Domain Mixup Architecture

Despite the reported high performance of deep learning models in crafted training data, generalization remains the challenge due to differences in publicly available and real-world clinical datasets. Our UDA method helps train models that can generalize between datasets, thereby significantly improving the results and lowering the cost of using deep learning models in clinical practice.

### 2.2.2 From Unsupervised to Semi-Supervised Domain Adaptation

One of the limitations of unsupervised domain adaptation settings is that we usually have a small number of labeled samples from the target domain in the real world. So how to make the most advantage of that becomes a vital problem. Few-shot learning approaches take advantage of that small portion of labeled target data but ignore a large number of unlabeled samples, while UDA methods do not take advantage of the small number of labeled samples from the target domain at all. In this section, we address this particular scenario as *Semi-Supervised Domain Adaptation* (SSDA), in which a relatively small amount of labeled data is available in the target domain.

In *Semi-Supervised Domain Adaptation* (SSDA), we have the access to the labeled source domain $\mathcal{S} = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ where $y_i^s \in \mathcal{Y} = \{1, ..., Y\}$ with $n_s$ annotated pairs. In the target domain, we are also given a limited number of labeled target samples $\mathcal{T}_l = \{(\mathbf{t}_i^t, y_i^t)\}_{i=1}^{n_t}$, as well as a relatively large number of unlabeled samples $\mathcal{T}_u = \{\mathbf{t}_j^u\}_{j=1}^{n_u}$. Our goal is to find a model that performs well on unlabeled target domain test data.

Domain adaptation approaches are effective at aligning source and target feature distributions without any labels from the target domain. In the real-world scenario, we usually have access to a limited number of labeled targets, which can help to improve the performance, but UDA frameworks do not take advantage of those [111, 61]. This problem is addressed as semi-supervised domain adaptation. SSDA is a vital task in computer vision and deep learning [2, 22, 136]. However, it is not sufficiently explored, especially concerning deep learning-based methods. MME [111] proposes a minimax entropy approach that adversarially optimizes an adaptive few-shot learning model, and the key idea is to minimize the distance between the class prototypes and neighboring unlabeled target samples. BiAT [61] proposes a bidirectional adversarial training method to effectively generate adversarial samples and bridge the domain gap. Both MME and BiAT focus on aligning source and target features by minimizing errors. However, Mixup [140] claims that neural networks trained to minimize errors change their predictions drastically when evaluated on examples just outside the training distribution. This makes the model fail to generalize well to adversarial samples and samples from other domains. ICT [125] proposes a Mixup approach to move the decision boundary to low-density regions of data distribution and achieves state-of-the-art performance on semi-supervised learning. An illustration of the Mixup-based method can be found in Fig 2.3. However, ICT focuses on semi-supervised learning and ignores the distribution shift between domains.

## 2.3 Self-Supervised Reconstruction

To exploit the temporal information in the input video sequences in the unlabeled target sets. and learn domain-invariant features from various targets, we follow the state-of-the-art self-supervised depth estimation work [49] and reconstruct the appearance of a target image from the viewpoint of an adjacent image by combining predicted depth, pose, and known camera intrinsic parameters. The process is illustrated in Fig. 2.4.

The pose regressor in our model yields the relative pose $T_{t\to t'}$ for each source view image $I_{t'}$, with respect to the target image $I_t$, from a consecutive monocular video sequence, by taking a pair of features extracted from $(I_t, I_{t'})$ as the inputs. The depth estimation decoder predicts a dense depth map $D_t$ simultaneously. Our goal is to minimize the reconstruction error $L_r$, where

$$L_r \quad = \quad \sum_{t'} ||I_t - I_{t'\to t}||. \tag{2.1}$$

The image reconstruction loss, in our case, is the $\ell_1$ distance in pixel space. By using the source image $I_{t'}$, the predicted depth $D_t$, the relative pose $T_{t\to t'}$, and the camera intrinsic parameters $K$, we can reconstruct the target image $I_t$ by:

$$I_{t'\to t} \quad = \quad I_{t'}\Big\langle proj(D_t, T_{t\to t'}, K)\Big\rangle. \tag{2.2}$$

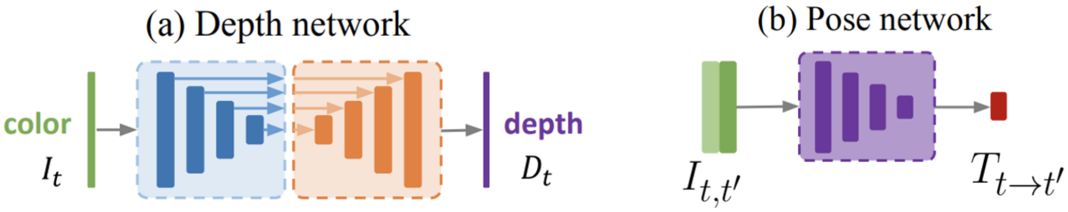


Figure 2.4: Self-Supervised Reconstruction Illustration [49].

where $proj()$ are the resulting 2D coordinates of the projected depths $D_t$ in $I_{t'}$ and $\langle\rangle$ is the sampling operator. To reduce noise in the prediction, we use edge-aware smoothness [48, 49]:

$$L_s = |\partial_x d_t^*|\, e^{-|\partial_x I_t|} + |\partial_y d_t^*|\, e^{-|\partial_y I_t|}, \tag{2.3}$$

where $d_t^* = d_t/\overline{d_t}$ is the mean-normalized inverse depth to discourage shrinking of the estimated depth. The complete self-supervised loss can be represented as:

$$\mathcal{L}'_{recon} = L_r + \lambda_s L_s \tag{2.4}$$

## 2.4 Prototypical Learning

Prototypical learning is a widely applied approach in deep learning, particularly in few-shot settings, which aims to learn a set of prototypes that effectively represent each class. The focus on few-shot segmentation has been growing in recent years, with various methods proposed to address the challenge. A conditional branch [115] was first proposed for few-shot segmentation to generate parameters $\theta$ from the support set for tuning the segmentation of the query set. A follow-up [104] combined extracted support features with the query features and utilized a decoder to produce the segmentation maps. A masked average pooling method [143] was proposed to enhance the extraction of foreground/background features from the support set. And an attention model [60] investigated guiding at multiple stages of the segmentation architecture. These methods generally adopt a parametric module, which blends information obtained from the support set to produce the segmentation results.

A metric learning-based method [25] tackled few-shot segmentation using the concept of prototypical learning networks. But the issue is that the approach is complex, requiring three training stages and intricate training configurations. Additionally, their method determined prototypes based on an image-level loss and utilized them to guide the segmentation of the query set, as opposed to directly obtaining segmentation from the metric learning. In contrast, our model features a simpler design, more akin to the Prototypical

Network [117]. Furthermore, it has been established [103] that late fusion is the optimal way to integrate the groundtruth annotations, making it more suitable for handling cases with sparse or changing masks.

My work in Chapter 5 is strongly connected to prototypical learning, and the goal is to develop a segmentation model that can quickly learn to perform segmentation based on a limited number of annotated images from new classes. Consistent with previous works [115], the following model training and testing protocols are adopted. Consider two sets of non-overlapping classes, $\mathcal{C}_{\text{seen}}$ and $\mathcal{C}_{\text{unseen}}$, from which the training set $\mathcal{D}_{\text{train}}$ is built using $\mathcal{C}_{\text{seen}}$ and the test set $\mathcal{D}_{\text{test}}$ is built using $\mathcal{C}_{\text{unseen}}$. The segmentation model $\mathcal{M}$ is trained on $\mathcal{D}_{\text{train}}$ and evaluated on $\mathcal{D}_{\text{test}}$. Our training and test sets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are structured as a series of *episodes*. Each episode includes a group of annotated support images, $\mathcal{S}$, and a set of query images, $\mathcal{Q}$. The training set is composed of $N_{\text{train}}$ episodes, represented as $\mathcal{D}_{\text{train}} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{N_{\text{train}}}$ while the test set consists of $N_{\text{test}}$ episodes, represented as $\mathcal{D}_{\text{test}} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{N_{\text{test}}}$. The task for each training/test episode $(\mathcal{S}_i, \mathcal{Q}_i)$ is defined as a $C$-way $K$-shot segmentation learning problem. The support set $\mathcal{S}_i$ includes $K$ $\langle$image, mask$\rangle$ pairs per semantic class and there are $C$ unique classes from $\mathcal{C}_{\text{seen}}$ for training and $\mathcal{C}_{\text{unseen}}$ for testing, represented as $\mathcal{S}_i = \{(I_{c,k}, M_{c,k})\}$ with $k = 1, 2, \cdots, K$ and $c \in \mathcal{C}_i$, where $|\mathcal{C}_i| = C$. The query set $\mathcal{Q}_i$ contains $N_{\text{query}}$ $\langle$image, mask$\rangle$ pairs from the same set of classes $\mathcal{C}_i$ as the support set. The model first extracts knowledge about the $C$ classes from the support set and then applies the learned knowledge to perform segmentation on the query set. As each episode contains different semantic classes, the model is trained to generalize well to unseen classes. After training the segmentation model $\mathcal{M}$ using $\mathcal{D}_{\text{train}}$, its performance on few-shot segmentation tasks is evaluated using $\mathcal{D}_{\text{test}}$. This evaluation involves utilizing the model to perform segmentation on the images in the query set $\mathcal{Q}_i$ for each testing episode, using the support set $\mathcal{S}_i$ as guidance.

By following this protocol, prototypical models learn representative and well-separated prototypes for each semantic class, using the prototypical network [117]. PANet [129] improves upon the previous approach of averaging over the entire input image [117] by using the groundtruth annotations in the support images to learn separate prototypes for the foreground and background. Two strategies for exploiting the segmentation masks exist, early fusion and late fusion [103]. Early fusion involves masking the support images before they are fed into the encoder [115, 60, 25], while late fusion masks the feature maps directly to create separate foreground/background features [143, 104]. Late fusion strategy was adopted in PANet [129] since it maintains consistency for the shared encoder network.

To obtain high-quality class prototypes, prototypical learning models [129, 117] first extract the feature map $F_{c,k}$ for the image $I_{c,k}$ from a given support set $\mathcal{S}_i = \{(I_{c,k}, M_{c,k})\}$, where $c$ represents the class index and $k = 1, \dots, K$ represents indices of the images in the support set. And the prototype of class $c$ is generated by using a masked average pooling module [143]:

$$p_c = \frac{1}{K} \sum_k \frac{\sum_{x,y} F_{c,k}^{(x,y)} 1[M_{c,k}^{(x,y)} = c]}{\sum_{x,y} 1[M_{c,k}^{(x,y)} = c]}, \tag{2.5}$$

where $(x, y)$ stands for the location of a pixel, and $1(\cdot)$ represents an indicator function that takes the value of $1$ if the $M_{c,k}^{(x,y)} = c$ is satisfied, and $0$ otherwise.

These prototypes are optimized in an end-to-end manner through non-parametric metric learning, as explained in Chapter 5.

Chapter 3

# Semi-Supervised Domain Adaptation for Image Classification

In this chapter, we introduce a semi-supervised domain adaptation framework for image classification. Most research on domain adaptation has focused on the purely unsupervised setting, where no labeled examples in the target domain are available. However, in many real-world scenarios, a small amount of labeled target data is available and can be used to improve adaptation. We address this semi-supervised setting and propose to use dynamic feature alignment to address both inter- and intra-domain discrepancy. Unlike previous approaches, which attempt to align source and target features within a mini-batch, we propose to align the target features to a set of dynamically updated class prototypes, which we use both for minimizing divergence and pseudo-labeling. By updating based on class prototypes, we avoid problems that arise in previous approaches due to class imbalances. Our approach, which doesn't require extensive tuning or adversarial training, significantly improves the state of the art for semi-supervised domain adaptation. We provide a quantitative evaluation on two standard datasets, DomainNet and Office-Home, and performance analysis.

## 3.1 Introduction

Deep neural networks have achieved impressive performance on a wide range of tasks, including image classification, semantic segmentation, and object detection. However, models often generalize poorly to new domains, such as when a model trained on indoor imagery is used to interpret an outdoor image. *Domain Adaptation* (DA) methods aim to take a model trained on a label-rich source domain and make it generalize well to a label-scarce target domain. Recently, most studies on domain adaptation have focused on the *Unsupervised Domain Adaptation* (UDA) setting, in which no labeled target data is available. However, in real-world scenarios, there is often a small amount of labeled target data available: the *Semi-Supervised Domain Adaptation* (SSDA) setting. Recent works [65, 111] demonstrate that directly applying UDA methods in the semi-supervised setting can actually hurt performance. Therefore, finding a way to effectively use the small amount of labeled target imagery is an important problem. We propose a novel approach that is tailored to the SSDA setting.

In addition to poor generalization in terms of model performance, the intermediate features for source and target domain inputs often display a significant domain shift. This motivates approaches that use feature alignment strategies [38, 9, 23, 24, 58, 152, 61, 111] to minimize the distances between source and target distributions. These methods address the shift between source and unlabeled target samples but ignore the shift within the target domain brought by the small number of labeled target samples. A recent work APE [65] addresses this issue as *intra-domain discrepancy* and proposes three schemes—*attraction, perturbation, and exploration*, to alleviate this discrepancy. *Attraction* is used to push the unlabeled features to the labeled feature distribution. *Perturbation* aims to move both labeled and unlabeled target features to their intermediate regions to minimize the gap in between. *Exploration* is complementary to the other two schemes by selectively aligning unlabeled target features to the source.

Due to the imbalance between the large amount of labeled source data and the small amount of labeled target data as well as the class imbalance (e.g. the existence of long-tail classes), a random mini-batch of aligned features can not always represent the true distribution of the data. Therefore, the alignment of unlabeled features can be inaccurate. Moreover, errors can be accumulated when incorrectly predicted unlabeled samples are selected to be used for pseudo-label training during *exploration*.

Considering the concerns mentioned above, we propose a novel *Dynamic Feature Alignment* (DFA) framework for the SSDA problem. Instead of directly aligning the unaligned target features to the aligned features within a mini-batch, we propose to align the unlabeled target features to a set of dynamically updated class prototypes, which are stored in a dynamic memory bank $\mathcal{B}$. To utilize these prototypes for pseudo-labeling, we selectively collect unlabeled samples based on their distances to class prototypes and their prediction entropy. We evaluate our method on standard domain adaptation benchmarks, including DomainNet and Office-Home, and results show that our method achieves significant improvement over the state of the art in both the 1-shot and 3-shot settings.

## 3.2 Related Works

We introduce related works in unsupervised and semi-supervised domain adaptation and describe their relationship to our approach.

### 3.2.1 Unsupervised Domain Adaptation

UDA is a machine learning technique that trains a model on one or more source domains and attempts to make the model generalize well on a different but related target domain [4, 106]. One of the key challenges of UDA is to mitigate the domain shift (or distribution shift) between the source and target domains. In general, three types of techniques can be used [130, 132]: (1) adversarial, (2) reconstruction based, and (3) divergence based.

The adversarial methods achieve domain adaptation by using adversarial training [24, 58, 152, 61, 111]. For instance, CoGAN [78] uses two generator/discriminator pairs for both the source and target domain, respectively, to generate synthetic data that is then used to train the target domain model. Domain-Adversarial Neural Networks (DANN) [39] promote the emergence of features that are discriminative on the source domain and unable to discriminate between the domains.

Reconstruction-based methods [45, 5, 44] use an auxiliary reconstruction task to create a representation that is shared by both domains. For instance, Deep Reconstruction Classification Network (DRCN) [45] jointly learns a shared encoding representation from two simultaneously running tasks. Domain Separation Networks (DSN) [5] propose a scale-invariant mean squared error reconstruction loss. Those learned representations preserve discriminability and encode useful information from the target domain.

While adversarial methods are often difficult to train and reconstruction-based methods require heavy computational cost, divergence-based methods align the domain distributions by minimizing a divergence that measures the distance between the distributions during training with minimal extra cost. For instance, Maximum Mean Discrepancy (MMD) [51] has been used in [110] to align the features of two domains by using a two-branch neural network with unshared weights. Deep CORAL [120] uses the Correlation Alignment (CORAL) [119] as the divergence measurement and Contrastive Adaptation Network (CAN) [64] measures the Contrastive Domain Discrepancy (CCD). Our proposed method DFA can be categorized as a divergence-based method.

### 3.2.2 Semi-Supervised Domain Adaptation

UDA approaches are effective at aligning source and target feature distributions without taking advantage of any labels from the target domain. In real-world scenarios, we usually have access to a small number of labeled target samples, which can be used to improve adaptation. This problem is addressed as semi-supervised domain adaptation (SSDA) [111, 61, 65, 90, 121]. Conventional UDA methods can be applied to SSDA simply by combining the source data with labeled target data. However, due to the imbalance between the large amount of source data and the small amount of labeled target data as well as the class imbalance issue, this strategy may align target features misleadingly [2, 22, 136].

To explore more effective solutions for the SSDA problem, BiAT [61] proposes a bidirectional adversarial training method to effectively generate adversarial samples and bridge the domain shift. MME [111] proposes a minimax entropy approach that adversarially optimizes an adaptive few-shot learning model. The key idea is to minimize the distance between the class prototypes and neighboring unlabeled target samples. The recent work APE [65] extends MME [111] by combining attraction, perturbation, and exploration strategies to bridge the intra-domain discrepancy. In contrast to previous works, we abandon the direct alignment between source and target features. Instead, our method is built upon a set of dynamically updated class prototypes.

### 3.2.3 Memory Bank

Memory banks have been applied in unsupervised learning and contrastive learning [133, 54, 79] as a dictionary look-up to reduce the computational complexity of calculating distances or similarities between features. For instance, the memory bank in [133] is designed to compute the non-parametric softmax classifier more efficiently for large datasets. To learn discriminative features from unlabeled samples, it stores one instance per class and is updated with the newly seen instances every iteration. The smoothness of the training was encouraged by adding a proximal regularization term, not a momentum update directly applied to the features. Another similar work is MoCo [54], which maintains a dynamic dictionary as a queue to replace old features with the current batch of features. Instead of applying the momentum on the representations, MoCo uses momentum to keep the encoder slowly evolving during training. The dictionary size of MoCo can be very large, but it is not designed to be class-balanced during training when the size is small. Both [133, 54] are effective for contrastive learning but not ideal for SSDA when the goal is to learn stable, representative, and class-balanced prototypes for aligning features from different domains and pseudo-labeling. Our work aims to resolve this issue by designing a dynamic memory bank for better feature alignment.

## 3.3 Approach

We introduce *Dynamic Feature Alignment* (DFA), a domain adaptation approach designed to work well in the semi-supervised setting. In the remainder of this section, we formalize the problem and describe the key components of our approach (see Fig 3.1 for an overview).

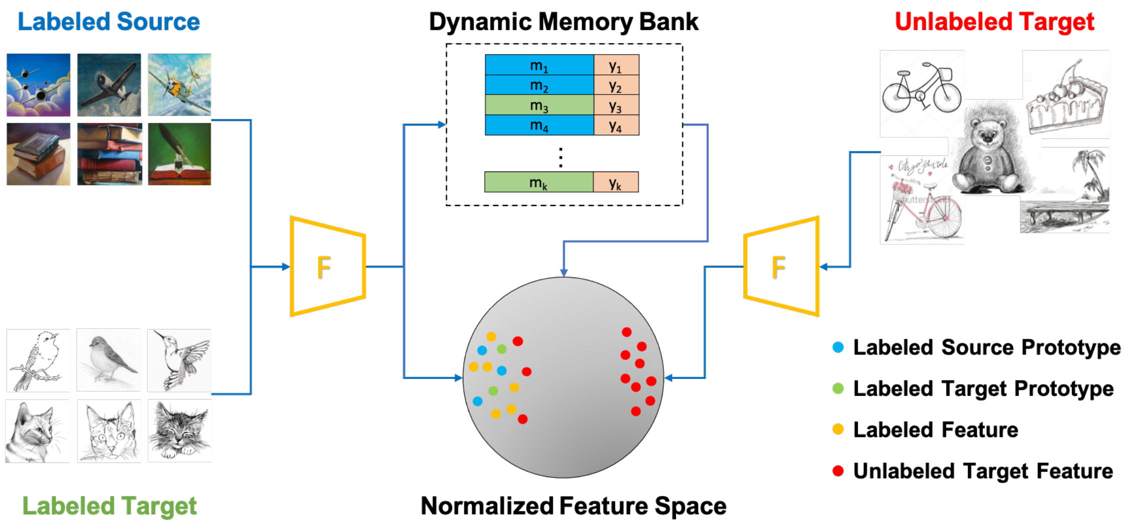


Figure 3.1: Framework of Dynamic Feature Alignment. Both labeled source and labeled target samples are passed into the feature extractor $F$. Normalized feature embeddings and labels are then stored in the dynamic memory bank as class prototypes for dynamic alignment and pseudo-labeling. The feature extractor for unlabeled target samples shares the same weights.

### 3.3.1 Problem Statement

In *SSDA*, we have the access to many labeled source domain samples $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ where $y_i^s \in \mathcal{Y} = \{1, ..., Y\}$ with $n_s$ annotated pairs. In the target domain, we are also given a limited number of labeled target samples $\mathcal{D}_l = \left\{\left(x_i^l, y_i{}^l\right)\right\}_{i=1}^{n_l}$, as well as a relatively large number of unlabeled samples $\mathcal{D}_u = \{(x_i^u)\}_{i=1}^{n_u}$. Our goal is to train a robust model that can perform well on the target domain by making the most advantage of $\mathcal{D}_s$, $\mathcal{D}_l$, and $\mathcal{D}_u$.

### 3.3.2 Supervised Classification in Normalized Feature Space

Our proposed DFA framework aligns the features of labeled examples from both domains using a supervised classification loss. A feature extractor $f(\cdot)$ is trained to extract the features from the input image $x$. The feature $m$, which is normalized to be unit length, is represented as:

$$m = f(x). \tag{3.1}$$

The classifier takes as inputs the normalized features $m$ and compares the cosine similarity between $m$ and the prototype weight vector $\mathbf{w}_k$ ($k$ = 1, ..., K) of class $k$. The similarities are scaled by a temperature $\tau = 0.05$, which controls the concentration level of the distribution [57, 127]. The probability of $m$ being categorized as class $k$ can be presented as:

$$P(k|m) = \frac{exp(\mathbf{w}^T \cdot m/\tau)}{\sum_{j=1}^{K} exp(\mathbf{w}_j^T \cdot m/\tau)}. \tag{3.2}$$

$P(k|m)$ is then passed to the classification loss [133, 54]. During training, we utilize cross-entropy as the supervised classification loss:

$$\mathcal{L}_{cls} = -\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_s\cup\mathcal{D}_l} \sum_{k=1}^{K} \log P_k. \tag{3.3}$$

When aligning the normalized features by training the network, the labeled target features $f(x^l)$ are closely aligned to the labeled source features $f(x^s)$ [111, 65], and we aim to utilize both labeled source and target samples and reduce the domain shift.

### 3.3.3 Dynamic Memory Bank

We propose to maintain a dynamic memory bank $\mathcal{B}$ that stores representative features, which we call prototypes, for each class. If our feature extractor was fixed, we could simply extract all the features once and compute their averages. This isn't feasible since we are actively updating our feature extractor. We could recompute the prototypes periodically, but this would be computationally expensive. Instead, we keep a weighted average of recently extracted features. A natural strategy for maintaining the memory bank would be to update the corresponding class prototype for each labeled sample in the current minibatch. This could be done, for example, as an exponentially weighted moving average (EWMA). The downside of this approach is that the class prototype for common classes would update more frequently, which leads to difficulty in setting update weights. It also doesn't take into account the potential for a large domain shift to lead to less informative class prototypes, especially when many samples are misclassified.

To maintain our dynamic memory bank $\mathcal{B}$, we propose to use an intermediate memory bank $b$ to enable us to make consistent, class-balanced updates. For every minibatch, we update $b$ as follows: we check the network output $f(x_k)$ on the input image $x_k$, which could be from the source or target domain. If $x_k$ is correctly classified, then the corresponding vector in $b$ is replaced with $f(x_k)$. Therefore, $b$ always stores the feature of the most recent correctly classified image for each class. We use the intermediate memory bank $b$ to update $\mathcal{B}$, using an EWMA, as follows:

$$\mathcal{B} = \gamma \cdot \mathcal{B}_{t-1} + (1-\gamma) \cdot b \tag{3.4}$$

where $\gamma$ regulates the pace of the update: a lower value results in a faster updating pace and a higher value leads to a slower update.

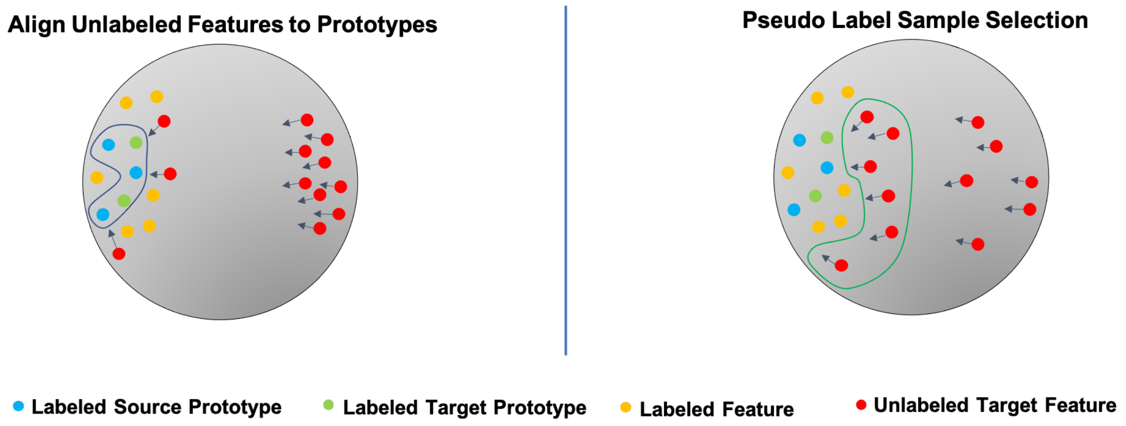

Align Unlabeled Features to Prototypes
Pseudo Label Sample Selection
● Labeled Source Prototype
● Labeled Target Prototype
● Labeled Feature
● Unlabeled Target Feature

Figure 3.2: Illustration of Dynamic Feature Alignment. (left) Shows that unlabeled target features (red dots) gradually move to class prototypes of the source (blue dots) and target (green dots). (right) Shows that unlabeled target features with lower entropy values and higher similarity scores (surrounded by the green curve) will be selected for pseudo-label training.

The dynamic memory bank $\mathcal{B}$ is updated based on labeled source features $f(x^s)$ and labeled target features $f(x^l)$. Therefore, each feature vector in $\mathcal{B}$ represents a class and can be interpreted as a class prototype, so $\mathcal{B}$ stores the prototypes of all classes. In the following section, we describe how this dynamic memory bank can be applied to (1) align the unlabeled target features to class prototypes accurately and (2) selectively collect unlabeled samples for pseudo-label estimation.

### 3.3.4 Dynamic Feature Alignment

Since the labeled target features $f(x^l)$ are already closely aligned to the labeled source features $f(x^s)$ [111, 65], we focus on the intra-domain discrepancy by directly minimizing the distance between $f(x^u)$ and class prototypes. We choose to use Maximum Mean Discrepancy (MMD) [52] to measure the difference between distributions. The basic idea of MMD is that if two distributions are identical, then all the statistics of these two should be the same [153].

$$D_{\mathcal{H}}(\mathcal{B},\mathcal{D}_u) \triangleq \left\|\mathbb{E}_{\mathcal{B}}[\phi(m_i)] - \mathbb{E}_{\mathcal{D}_u}[\phi(f(x_j^u))]\right\|_{\mathcal{H}}^2, \tag{3.5}$$

where $m_i$ is the $i$-th feature stored in $\mathcal{B}$, representing the prototype of class $i$, and $f(x_j^u)$ is the $j$-th feature in the unlabeled target features $\mathcal{D}_u$. Here $\phi(\cdot)$ represents Gaussian Radial Basis Function (RBF) kernels, which map the input feature maps to the reproducing kernel Hilbert space $\mathcal{H}$. The MMD loss can be presented as:

$$\mathcal{L}_{mmd} = D_{\mathcal{H}}(\mathcal{B},\mathcal{D}_u). \tag{3.6}$$

By minimizing $\mathcal{L}_{mmd}$, the domain discrepancy will be bridged and unaligned target features will be gradually moving close to the aligned class prototypes. See Fig 3.2 (left) for illustration.

To accelerate the learning process as well as further improve the accuracy, we take advantage of the large number of unlabeled samples and propose a pseudo-label estimation strategy. For an unlabeled image $x^u$, we compute the distances between $f(x^u)$ and every class prototype $m$ stored in $\mathcal{B}$. Here, cosine similarity is used for distance measurement. Since $m_i$ is the representative prototype of the $i$-th class, a higher similarity value represents a higher probability that $x^u$ belongs to class $i$. Then the pseudo-label estimation function can be formulated by using a softmax function with a temperature $\tau_p$ as follows:

$$P_{dist}(i|x^u) = \frac{exp(m^T \cdot f(x^u)/\tau_p)}{\sum_{j=1}^{K} exp(m_j^T \cdot f(x^u)/\tau_p)}. \tag{3.7}$$

Training with inaccurate pseudo-labels can accumulate errors. We adopt a sample selection strategy to keep the high-quality pseudo-labels in the training loop and eliminate the inaccurate ones. First, those samples with similarity scores higher than the threshold $\epsilon_{dist}$ will be stored in $M_{dist}$. Second, we collect samples with the prediction entropy $H_{\mathbf{w}}(P_{dist})$ less than a threshold $\epsilon_{ent}$ and store them in $M_{ent}$. This step will selectively collect the samples that are close to the aligned features [65]. See Fig 3.1 (right) for illustration. Last, we take the intersection of $M_{dist}$ and $M_{ent}$, noted as $M_{pse}$. Thus, only samples satisfying both conditions will be used in the pseudo-label training loop.

The network output $\hat{y}_x$ of each sample $x$ in $M_{pse}$ is used as the pseudo-label for calculating the cross entropy loss $\mathcal{L}_{pseudo}$:

$$\mathcal{L}_{pseudo} = -\mathbb{E}_{\mathcal{D}_u}[\mathbf{1}_{M_{pse}}(x) \log p(y = \hat{y}_x|x)]. \tag{3.8}$$

To further minimize the intra-domain discrepancy, we follow [65] and apply the same perturbation scheme. We regularize the perturbed features and the raw, clean features by Kullback–Leibler divergence. The goal is to enforce the model to generate perturbation-invariant features so that the perturbation loss can be presented as:

$$\mathcal{L}_{perturb} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_u \cup \mathcal{D}_l}\left[\sum_{i=1}^{K} KL[f(x), f(x + r_x)]\right], \tag{3.9}$$

where $x$ is the input image, and $r_x$ is the optimized perturbation added to $x$.

### 3.3.5 Overall Loss Function

The overall loss function of the proposed DFA framework is the weighted sum of every different piece of the loss function mentioned above and can be integrated as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha_1 \mathcal{L}_{mmd} + \alpha_2 \mathcal{L}_{pseudo} + \alpha_3 \mathcal{L}_{perturb}. \tag{3.10}$$

## 3.4 Evaluation

We evaluate our method by conducting experiments on two standard domain adaptation benchmarks. Below we describe the datasets used for these experiments, implementation details, and extensive performance analysis.

### 3.4.1 Datasets

DomainNet [101] is a large-scale domain adaptation benchmark that contains 6 domains and 345 object categories. Following the previous works MME [111] and APE [65], we use a 4 domain (Real, Clipart, Painting, Sketch) subset with 126 classes. We report our results on 7 scenarios for a fair comparison with the previous state-of-the-art works.

We also evaluate our model on Office-Home [124], which is another domain adaptation benchmark that contains 4 domains (Real, Clipart, Art, Product) and 65 categories. Here we report results on all 12 adaptation scenarios.

### 3.4.2 Implementation Details

We implement our model using PyTorch [100]. We follow [111, 65] and report results on DomainNet using AlexNet and ResNet-34 as the backbones. For Office-Home, we report on ResNet-34. All networks are pre-trained on ImageNet. Our model is trained on labeled $\mathcal{D}_s$, $\mathcal{D}_l$, and unlabeled $\mathcal{D}_u$. To make the source and target balanced in the training stage, each mini-batch of labeled samples contains half source samples and half target samples. We consider both one-shot and three-shot settings, and for each class, one (or three) labeled target samples are given for training. For evaluation, we reveal the ground-truth labels of $\mathcal{D}_u$ and report results on that. We follow [111, 65] and use SGD optimizer with an initial learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.0005. For hyperparameters, we set the temperature for the classifier as $\tau = 0.05$, and set $\gamma = 0.1$ to update $\mathcal{B}$ at a fast pace. We set the temperature for pseudo-label estimation as $\tau_p = 0.07$. As for thresholds, we set $\epsilon_{dist}$ to $0.3$ for ResNet-34 and $0.1$ for AlexNet and set $\epsilon_{ent} = 0.5$ for both backbone networks.

Table 3.1: Classification accuracy (%) on the DomainNet dataset for three-shot setting with 4 domains, 7 scenarios using AlexNet and ResNet-34 as backbone networks, respectively.

| **Net** | **Method** | R to C | R to P | P to C | C to S | S to P | R to S | P to R | MEAN |
|---|---|---|---|---|---|---|---|---|---|
| AlexNet | S+T | 47.1 | 45.0 | 44.9 | 36.4 | 38.4 | 33.3 | 58.7 | 43.4 |
| | DANN | 46.1 | 43.8 | 41.0 | 36.5 | 38.9 | 33.4 | 57.3 | 42.4 |
| | ADR | 46.2 | 44.4 | 43.6 | 36.4 | 38.9 | 32.4 | 57.3 | 42.7 |
| | CDAN | 46.8 | 45.0 | 42.3 | 29.5 | 33.7 | 31.3 | 58.7 | 41.0 |
| | ENT | 45.5 | 42.6 | 40.4 | 31.1 | 29.6 | 29.6 | 60.0 | 39.8 |
| | MME | 55.6 | 49.0 | 51.7 | 39.4 | **43.0** | 37.9 | 60.7 | 48.2 |
| | SagNet | 49.1 | 46.7 | 46.3 | 39.4 | 39.8 | 37.5 | 57.0 | 45.1 |
| | APE | 54.6 | 50.5 | **52.1** | 42.6 | 42.2 | 38.7 | 61.4 | 48.9 |
| | Ours | **55.0** | **52.3** | 51.6 | **44.5** | 41.8 | **39.4** | **62.1** | **49.5** |
| ResNet | S+T | 60.0 | 62.2 | 59.4 | 55.0 | 59.5 | 50.1 | 73.9 | 60.0 |
| | DANN | 59.8 | 62.8 | 59.6 | 55.4 | 59.9 | 54.9 | 72.2 | 60.7 |
| | ADR | 60.7 | 61.9 | 60.7 | 54.4 | 59.9 | 51.1 | 74.2 | 60.4 |
| | CDAN | 69.0 | 67.3 | 68.4 | 57.8 | 65.3 | 59.0 | 78.5 | 66.5 |
| | ENT | 71.0 | 69.2 | 71.1 | 60.0 | 62.1 | 61.1 | 78.6 | 67.6 |
| | MME | 72.2 | 69.7 | 71.7 | 61.8 | 66.8 | 61.9 | 78.5 | 68.9 |
| | SagNet | 62.0 | 62.9 | 61.5 | 57.1 | 59.0 | 54.4 | 73.4 | 61.5 |
| | APE | 76.6 | 72.1 | **76.7** | 63.1 | 66.1 | **67.5** | 79.4 | 71.7 |
| | Ours | **76.7** | **73.9** | 75.4 | **65.5** | **70.5** | **67.5** | **80.3** | **72.8** |

Table 3.2: Classification accuracy (%) on the DomainNet dataset for one-shot setting with 4 domains, 7 scenarios using ResNet-34.

| **Net** | **Method** | R to C | R to P | P to C | C to S | S to P | R to S | P to R | MEAN |
|---|---|---|---|---|---|---|---|---|---|
| ResNet | S+T | 55.6 | 60.6 | 56.8 | 50.8 | 56.0 | 46.3 | 71.8 | 56.9 |
| | DANN | 58.2 | 61.4 | 56.3 | 52.8 | 57.4 | 52.2 | 70.3 | 58.4 |
| | ADR | 57.1 | 61.3 | 57.0 | 51.0 | 56.0 | 49.0 | 72.0 | 57.6 |
| | CDAN | 65.0 | 64.9 | 63.7 | 53.1 | 63.4 | 54.5 | 73.2 | 62.5 |
| | ENT | 65.2 | 65.9 | 65.4 | 54.6 | 59.7 | 52.1 | 75.0 | 62.6 |
| | MME | 70.0 | 67.7 | 69.0 | 56.3 | 64.8 | 61.0 | 76.1 | 66.4 |
| | SagNet | 59.4 | 61.9 | 59.1 | 54.0 | 56.6 | 49.7 | 72.2 | 59.0 |
| | APE | 70.4 | 70.8 | **72.9** | 56.7 | 64.5 | **63.0** | 76.6 | 67.6 |
| | Ours | **71.8** | **72.7** | 69.8 | **60.8** | **68.0** | 62.3 | **76.8** | **68.9** |

### 3.4.3 Baselines

We compare our proposed method with different models. Baselines include training a network only using labeled samples (S+T), an entropy minimization-based semi-supervised method (ENT [50]), three feature alignment-based unsupervised domain adaptation models (DANN [38], ADR [112], and CDAN [81]), and three state-of-the-art semi-supervised learning models (SagNet [91], MME [111], and APE [65]) that aim to the same goal as our method.

Table 3.3: Classification accuracy (%) comparisons on Office-Home for three-shot setting with 4 domains, 12 scenarios using ResNet-34 as the backbone network.

| **Method** | R to C | R to P | R to A | P to R | P to C | P to A | A to P | A to C | A to R | C to R | C to A | C to P | MEAN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S+T | 55.7 | 80.8 | 67.8 | 73.1 | 53.8 | 63.5 | 73.1 | 54.0 | 74.2 | 68.3 | 57.6 | 72.3 | 66.2 |
| DANN | 57.3 | 75.5 | 65.2 | 69.2 | 51.8 | 56.6 | 68.3 | 54.7 | 73.8 | 67.1 | 55.1 | 67.5 | 63.5 |
| CDAN | 61.4 | 80.7 | 67.1 | 76.8 | 58.1 | 61.4 | 74.1 | 59.2 | 74.1 | 70.7 | 60.5 | 74.5 | 68.2 |
| ENT | 62.6 | 85.7 | 70.2 | 79.9 | 60.5 | 63.9 | 79.5 | 62.3 | 79.1 | 76.4 | 64.7 | 79.1 | 71.9 |
| MME | 64.6 | 85.5 | 71.3 | 80.1 | 64.6 | 65.5 | 79.0 | 63.6 | 79.7 | 76.6 | 67.2 | 79.3 | 73.1 |
| APE | 66.4 | 86.2 | 73.4 | 82.0 | 65.2 | 66.1 | **81.1** | **63.9** | 80.2 | **76.8** | 66.6 | **79.9** | 74.0 |
| Ours | **68.3** | **86.9** | **74.1** | **82.3** | **65.9** | **67.8** | 80.4 | 63.0 | **80.3** | 76.6 | **67.8** | 79.1 | **74.4** |

### 3.4.4 Experiment Results

We summarize the comparisons between our method and the baselines on 7 adaptation scenarios of the DomainNet dataset in Table 3.1 and Table 3.2, for three-shot setting and one-shot setting respectively. When using the ResNet-34 as the backbone network, our method outperforms the current state-of-the-art baseline by more than $1\%$ and achieves the best performance in most adaptation scenarios. For the best case *S to P* of the three-shot setting, our method surpasses the second-best method by $4.4\%$. When AlexNet is applied as the backbone network, the margin that our method outperforms other methods is not as large as using ResNet-34 because our proposed scheme requires high-quality intermediate features stored in $\mathcal{B}$, and ResNet-34 has more advantage in achieving that compared with AlexNet. Our method still performs the best in most scenarios, surpassing APE by $0.6\%$ on average.

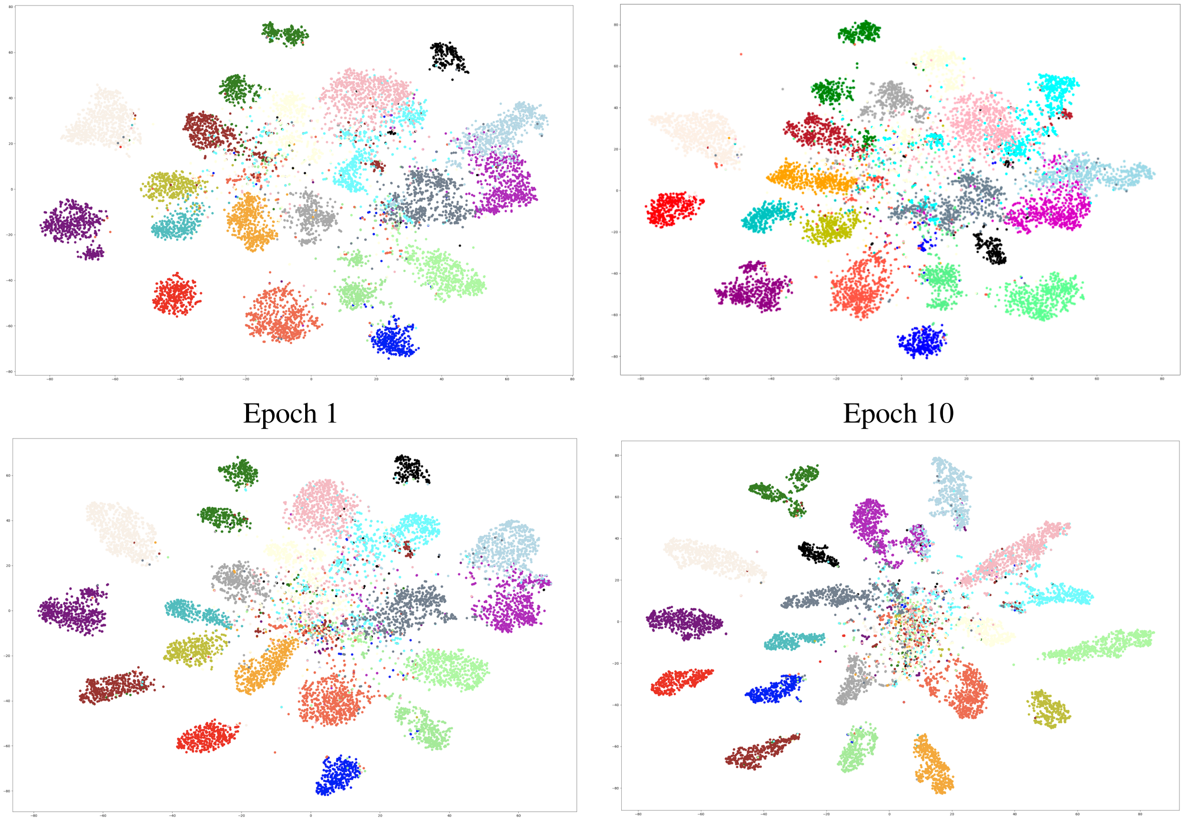
The comparison results of our method with other baselines of 12 adaptation scenarios on the Office-Home dataset are summarized in Table 3.3. We report results using ResNet-34 as the backbone network. Our method achieves the best performance on 8 out of 12 adaptation scenarios and outperforms all the baselines on average.

### 3.4.5 Analysis

In Sec 3.3.3, we explain the updating rules of the dynamic memory bank $\mathcal{B}$. Here we conduct an experiment to show how $\gamma$ affects the classification accuracy. We use AlexNet as the backbone network and train the model on DomainNet with different $\gamma$. A smaller $\gamma$ means updating $\mathcal{B}$ faster, and larger $\gamma$ means updating $\mathcal{B}$ in a more stable way. We evaluate 3 adaptation scenarios and summarize the results in Table 3.4. It demonstrates that a more stable $\mathcal{B}$ (with larger $\gamma$ as 0.75) can not help improve the performance, and replacing the entire $\mathcal{B}$ with the new one every iteration ($\gamma = 0$) can not get the optimal performance either. Our results show that using a small (but larger than $0$) $\gamma$ and updating

Table 3.4: Analysis on how $\gamma$ affects the classification accuracy (%).

| **Method** | R to C | R to P | C to S | MEAN |
|---|---|---|---|---|
| APE | 54.6 | 50.5 | 42.6 | 49.2 |
| Ours ($\gamma = 0.75$) | 54.1 | 50.7 | 42.4 | 49.0 |
| Ours ($\gamma = 0.25$) | 54.9 | 52.2 | 43.6 | 50.2 |
| Ours ($\gamma = 0.1$) | **55.0** | **52.3** | **44.5** | **50.6** |
| Ours ($\gamma = 0$) | 54.7 | 52.0 | **44.5** | 50.4 |



Epoch 1

Epoch 10

Epoch 50

Epoch 100

Figure 3.3: The t-SNE visualization of intermediate features in the target domain of our method at different training stages.

$\mathcal{B}$ at a relatively faster pace achieves the best classification accuracy.

To better understand the feature alignment progress, we show the t-SNE [123] embedding of the intermediate features at different training stages in Fig 3.3. We visualize the target features extracted by ResNet-34 in the experiment of *Painting to Real* scenario of the DomainNet. Following APE [65], we randomly select 20 classes out of 126 classes in the dataset for clarity. This shows that as training progresses, the target feature clusters will gradually be split for better classification.

### 3.5 Conclusion

We proposed a novel approach for semi-supervised domain adaptation that uses a dynamic memory bank to support inter- and intra-domain feature alignment. Our update approach is designed to be class balanced, thereby mitigating one of the more challenging aspects of the problem. We evaluated our approach on two standard datasets and found that it significantly improved the average accuracy over the previous state-of-the-art techniques. In addition to improved accuracy, our approach has several attractive features. It doesn't require significant additional memory (only two copies of the class prototypes) or computation (only online updates to the intermediate and dynamic memory banks). It also doesn't require extensive parameter tuning: the weights for the loss function are fixed across all experiments, accuracy isn't particularly sensitive to the dynamic memory updates parameter $\gamma$, and the pseudo-label thresholds only needed to be adjusted to account for the low discriminative power of AlexNet. Given this, we believe this approach will be applicable to many semi-supervised domain adaptation scenarios.

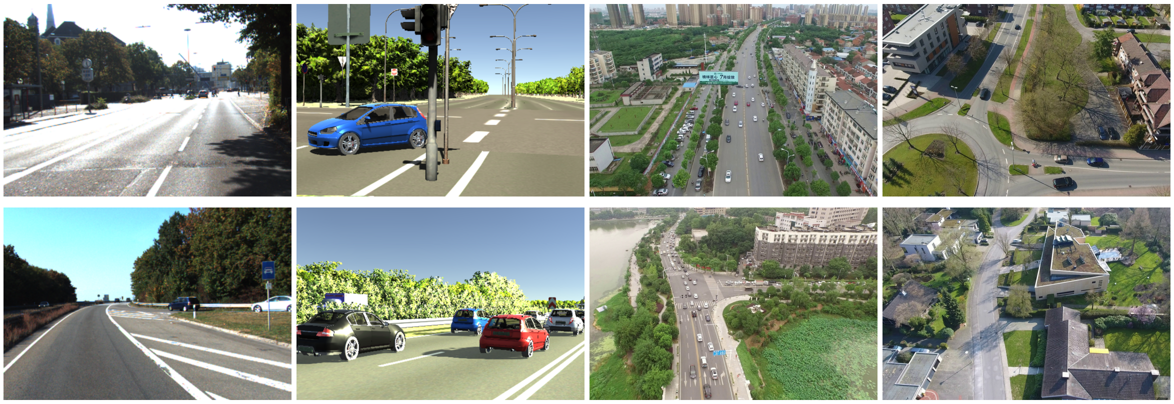
Chapter 4

# CrossDepth: Cross-Scene Adaptation for Multi-Domain Depth Estimation

In this chapter, we introduce a cross-scene depth estimation framework under the multi-domain setting. We address the task of monocular depth estimation in the multi-domain setting. Given a large dataset (source) with ground-truth depth maps, and a set of unlabeled datasets (targets), our goal is to create a model that works well on unlabeled target datasets across different scenes. This is a challenging problem when there is a significant domain shift, often resulting in poor performance on the target datasets. We propose to address this task with a unified approach that includes adversarial knowledge distillation and uncertainty-guided self-supervised reconstruction. We provide both quantitative and qualitative evaluations on four datasets: KITTI, Virtual KITTI, UAVid China, and UAVid Germany. These datasets contain widely varying viewpoints, including ground-level and overhead perspectives, which is more challenging than is typically considered in prior work on domain adaptation for single-image depth. Our approach significantly improves upon conventional domain adaptation baselines and does not require additional memory as the number of target sets increases.

## 4.1 Introduction

Deep neural networks have achieved impressive performance on a wide range of tasks, including image classification, semantic segmentation, and object detection. However, models often generalize poorly to new domains, such as when a model trained on synthetic imagery is used to process real-world imagery. *Domain Adaptation* (DA) methods aim to solve this problem by adapting a model trained on a label-rich source domain to a label-scarce target domain. Recently, most studies on domain adaptation have focused on the single target-domain setting [58, 82, 97], in which only one target domain is considered at a time. However, in many real-world scenarios, test data may be collected from various sources and domains. For instance, ground-level images collected by the same self-driving car can still be considered distinct domains due to different sensors, weather conditions,



KITTI Virtual KITTI UAVid China UAVid Germany

Figure 4.1: From left to right, we show examples of the four diverse datasets we propose to use, including a real-world ground-level dataset KITTI [41], a synthetic ground-level dataset Virtual KITTI [8], and two aerial imagery datasets UAVid China and UAVid Germany [85].

and changing environments. Recent work [113, 93] has demonstrated that directly applying conventional DA methods to the multi-target setting may not achieve optimal performance.

Most existing DA for single-image depth works [82, 70, 148] have been evaluated exclusively on ground-level indoor and outdoor images (e.g., adapting from synthetic-to-real or rainy-to-sunny). While these are important and challenging problems, this line of research has ignored the problem of domain adaptation across extreme viewpoint shifts. With the increasing prevalence of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), the need to adapt networks across such viewpoint shifts is increasingly important. Therefore, we consider the problem of domain adaptation between videos collected from UAVs and UGVs.

For both UAVs and UGVs, monocular depth estimation is an important fundamental task for many vision applications including localization, navigation, and scene recognition. However, obtaining ground-truth depth annotations is difficult, often requiring expensive LiDAR sensors. Based on this fact, adapting models trained on label-rich source datasets to label-scarce target datasets is meaningful for the task of depth estimation. Therefore, developing a method that can effectively adapt one source-trained model to multiple target datasets is important for both domain adaptation and monocular depth estimation tasks, and we propose a novel approach that tackles this problem.

Given a source dataset, with sufficient ground-truth depth, we can easily train a network for monocular depth estimation. Unfortunately, this performs poorly when applied out of the domain, especially when the network is trained on UGV imagery and applied to UAV imagery. We propose to address this task with a unified approach that includes adversarial knowledge distillation and uncertainty-guided self-supervised reconstruction.

When networks are applied to different domains, the intermediate feature distributions are often very different [23, 111]. This observation has motivated approaches to domain adaptation that focus on the learning domain-invariant features [38, 9, 23, 111]. One common solution is adversarial training, which uses a discriminator and an adversarial loss to encourage the encoder to learn domain-invariant features. However, conventional adversarial approaches are difficult to scale up when multiple targets are included. Recent work MTKT [114] proposes a multi-target adversarial training method to tackle this problem. However, MTKT [113] requires using a separate target decoder for each new target set to handle the instability problem. We propose to use a unified student-teacher model to distill knowledge from all source-target pairs with the guidance of a self-supervised reconstruction loss, without requiring additional memory.

To exploit the temporal information stored in the video sequences, we propose an uncertainty-guided self-supervised reconstruction module and apply it to the unlabeled target imagery. This requires both depth estimates and relative pose estimates. Therefore, in addition to depth estimation, our network is also trained to predict the relative pose between two adjacent frames. This reconstruction loss does not require ground-truth depths or camera poses, making it easy to apply to new target domains. To further improve the reliability of the self-supervised reconstruction, we estimate the uncertainty map by computing the average reconstruction error map from four adjacent frames in the video sequence. Pixels with higher uncertainty will be down-weighted in the reconstruction loss during training.

We evaluate our method on four datasets (see Fig. 4.1 for example imagery): KITTI, Virtual KITTI, UAVid China, and UAVid Germany. The evaluation shows that our method consistently outperforms several strong baseline methods.

Our key contributions are summarized as follows:

- We develop an adversarial knowledge distillation framework that can bridge the domain gaps between the source and multiple targets without requiring additional memory as the number of targets increases.

- We propose an uncertainty-guided, self-supervised reconstruction loss that can be easily applied to unlabeled new domains.

- We conduct extensive experiments on diverse datasets, which include real, synthetic, ground-level, and aerial images, and demonstrate that our model significantly reduces issues due to extreme domain shifts.

## 4.2 Related Work

We introduce related works in conventional domain adaptation, depth estimation, and multi-target domain adaptation and describe their relationship to our approach.

### 4.2.1 Domain Adaptation

The objective of domain adaptation (DA) is to train a model on one or more source domains and make the model generalize well to different but related target domains [4, 106]. One of the key challenges of DA is to mitigate the distribution shift between different domains. In general, three types of techniques have been explored [130, 132]: (1) divergence-based, (2) reconstruction-based, and (3) adversarial.

Divergence-based methods align the intermediate features by minimizing a divergence between the distributions during training. For instance, Maximum Mean Discrepancy (MMD) [51] has been used in a recent work [110] to align the features of two domains by using a two-branch neural network with unshared weights. Deep CORAL [120] uses Correlation Alignment (CORAL) [119] as the divergence measurement and Contrastive Adaptation Network (CAN) [64] measures the contrastive domain discrepancy.

Reconstruction-based methods [45, 5, 44] use an auxiliary reconstruction task to create a representation that is shared by both domains. For instance, Deep Reconstruction Classification Network (DRCN) [45] jointly learns a shared encoding representation from two simultaneously running tasks. DRANet [72] combines both reconstruction-based and adversarial methods to transfer visual attributes in latent space for domain adaptation.

The adversarial methods bridge the domain gaps by performing adversarial training [24, 58, 152, 61, 111]. For instance, CoGAN [78] uses two generator-discriminator modules for both the source and target domain, respectively, to synthesize realistic data that is then used to train the target domain model. Multi-scale adversarial domain adaptation module [73] is also used for domain adaptation for animal pose estimation. Our proposed method can be categorized into this type.

### 4.2.2 Monocular Depth Estimation

Recent research on monocular depth estimation can be categorized into three groups [63]: supervised, weakly supervised, and self-supervised. Supervised depth estimation networks [26, 27, 34] require a larger volume of ground-truth depth annotations. These methods formulate depth estimation as a regression problem and directly learn from the supervised losses. The weakly-supervised line of depth estimation works [82, 70] do not require depth annotations but require other labels, including semantic labels or odometry.

For instance, DESC [82] proposes an unsupervised domain adaptation depth estimation network that uses ground-truth semantic labels from both source and target domains to enforce the consistency between the predictions from a semantic branch and a depth estimation branch. CoMoDA [70] adds a velocity loss to Monodepth2 [49] and performs inference-time adaption to unseen test data. Pseudo-labeling-based methods [154, 15, 134] generate pseudo labels from internet photo collections by using the ground-truth ordinal depth information as a cue and leveraging multi-view stereo reconstruction algorithms. The self-supervised group explores learning algorithms using either rectified stereo image pairs [40, 48] or monocular video sequences [151, 137, 49, 63] as training data. The video-based depth estimation methods [77, 84] use consecutive monocular frames to estimate depth during inference, making the assumption that scenes are mostly rigid. There are also several works on monocular depth completion [29, 87, 135] that have been proposed to capitalize on sparse depth maps with corresponding images, resulting in dense depth estimations. Our proposed method uses monocular video sequences as training data and can be considered as a combination of supervised and self-supervised approaches.

Monocular depth estimation methods typically consider a single domain, usually ground-level indoor and outdoor images, without considering how they generalize to other domains, such as aerial images. A recent work [88] directly applies a variant of Monodepth2 [49] to UAV videos and achieves reasonable results. However, this work is also limited to a single domain of aerial images, and it does not consider adapting the model to both ground-level and aerial images simultaneously.

### 4.2.3 Multi-Target Domain Adaptation

Multi-target domain adaptation (MTDA) is a variant of domain adaptation. Most previous works in this area have focused on classification and semantic segmentation. There are two sub-settings of MTDA [114]. The first setting assumes that the domain labels are unknown during both training and testing [102, 16]. DAL [102] proposes an architecture that extracts domain-invariant features by performing source-target domain disentanglement and removing irrelevant features by adding a class disentanglement loss. BTDA [16] presents an adversarial meta-adaptation network that both aligns the source with mixed-target features and clusters the target inputs into $k$ adversarially aligned clusters by training an unsupervised meta-learner. The second setting assumes the domain labels of training samples are known during training but remain unknown during the inference stage. To handle this, ITA [46] jointly learns a domain classifier and a class label predictor to separately capture both domain-specific features and domain-invariant features. The recent work on MTDA classification [93] adopts an end-to-end multi-target network by using a gradient reversal

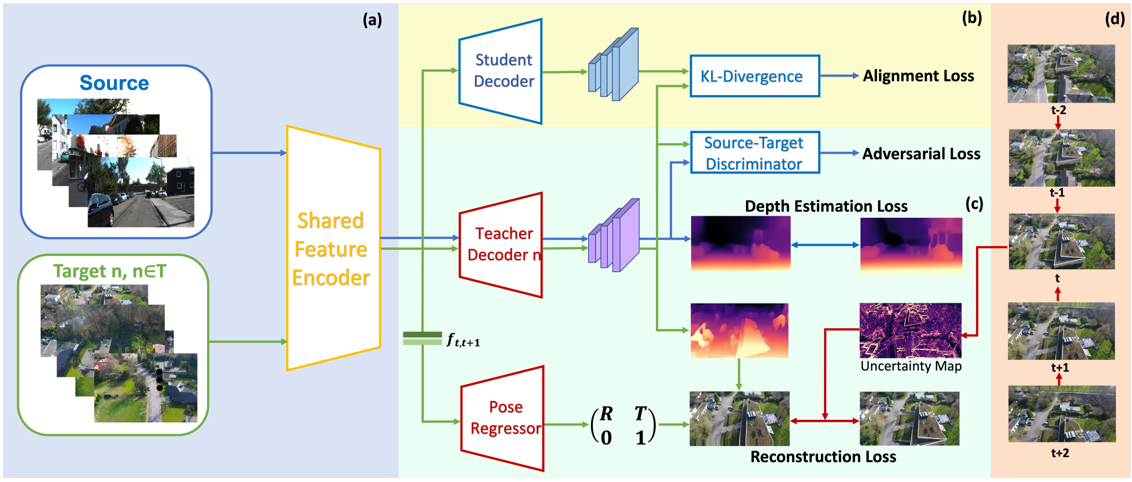


Figure 4.2: Overview of CrossAdapt. (a) Monocular video sequences from both source and target domains are passed into the shared feature encoder. (b) The student and teacher decoders are aligned by minimizing the KL divergence. (c) A teacher decoder takes features from both domains and estimates depth maps. The depth map of a target frame is combined with the relative pose predicted by the pose regressor to compute the reconstruction loss. The outputs of the teacher decoder are passed into a discriminator to compute the adversarial loss. (d) By taking the average of the reconstruction error maps of $t$ and its 4 adjacent frames, an uncertainty map is estimated to further guide the self-supervised reconstruction loss.

layer. MTKT [113] proposes a multi-target adversarial training framework for semantic segmentation. It uses multiple target-specific decoders to bridge the domain gaps caused by different target sets. Our work is similar to MTKT [113], but instead of considering every input image separately, our input is a sequence of video frames so more adjacent information can be exploited to boost the performance.

## 4.3 Approach

We introduce *Cross-Scene Adaptation for Multi-Domain Depth Estimation* (CrossAdapt), an approach for training a monocular depth-estimation network in the MTDA setting. Our work focuses on monocular videos because they are readily available, but our approach could be easily adapted to stereo pairs. In this section, we formalize the problem and describe the key components of our approach.

### 4.3.1 Problem Statement

We are given a set of fully labeled source-domain samples $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ where $x_i^s \in \mathbb{R}^{H \times W \times 3}$ represents an image in the source domain and $y_i^s \in \mathbb{R}^{H \times W}$ the corresponding ground-truth depth map. In addition, we are given $T$ sets of unlabeled samples $\mathcal{D}_{t,n} = \{x_i^{t,n}\}_{i=1}^{n_t}$, where $x_i^{t,n} \in \mathbb{R}^{H\times W\times 3}$ represents an image from the $n$-th target domain ($n \leq T$). Our goal is to train a robust monocular depth estimation model that can perform well on all of the target domains. This will require combining supervised training for depth estimation on the source domain and domain adaptation strategies capable of using the unlabeled target data.

### 4.3.2 Approach Overview

We visualize our overall network architecture in Fig. 4.2. During training, monocular video sequences from the source and target domains are passed into the shared encoder $E$. The encoded features from both source and target are then passed into a depth estimation teacher decoder $D_T$, which yields estimated depth maps for both inputs, and the source-target discriminator, which encourages the network to learn domain-invariant features. In addition, temporally adjacent target frames are passed into a pose regressor $D_P$ and a relative camera pose is estimated. The estimated pose and depth are combined, along with the known camera intrinsic parameters and the uncertainty map, to compute the reconstruction loss. We then minimize the KL divergence between the predictions of the student decoder $D_S$ and the teacher decoder $D_T$. This student decoder $D_S$ is the final model to be used for evaluation. Our model is trained towards four objectives: supervised depth estimation, adversarial loss, alignment loss, and uncertainty-guided reconstruction loss. We describe them in the following sections.

### 4.3.3 Supervised Depth Estimation

For source imagery, where ground-truth depth is available, our proposed CrossAdapt framework is trained in a supervised manner using a depth estimation loss. An encoder is trained to extract features from the input image $x$. The decoder takes as inputs the features and predicts the depth map $y^s$.

Here, we minimize the $\ell_2$ distance between the predicted depth $\tilde{y}^s$ and the ground-truth depth $y^s$:

$$\mathcal{L}_{supervised} = ||y^s - \tilde{y}^s||^2. \tag{4.1}$$

### 4.3.4 Adversarial Knowledge Distillation

One of the key goals of domain adaptation is to encourage the network to learn domain-invariant features. To achieve that goal, we use a source-target discriminator $D_D$ to classify the output feature $F_T$ that comes from the penultimate layer of $D_T$ as either source (1) or target (0) by using binary cross entropy $L_{BCE}$.

$$\mathcal{L}_{dis} = L_{BCE}(F_T, 1)_{source} + L_{BCE}(F_T, 0)_{target}. \tag{4.2}$$

Our network has two goals: one is to predict accurate depth maps, and the other one is to fool the discriminator. To achieve the second goal, here we use the response from the discriminator in the following loss and always encourage it to predict source (1) for all inputs. Note that the input feature to the discriminator comes from the penultimate layer of the teacher decoder.

$$\mathcal{L}_{extractor} = L_{BCE}(D_D(F_T), 1). \tag{4.3}$$

And the total adversarial loss is represented as:

$$\mathcal{L}_{adv} = \mathcal{L}_{dis} + \lambda_{adv}\mathcal{L}_{extractor}. \tag{4.4}$$

To distill knowledge from the teacher decoder the to student decoder, we use the output of the penultimate layer the teacher decoder $F_T$, and the student decoder $F_S$ to compute the KL divergence.

$$\mathcal{L}_{align} = \mathcal{L}_{KL}(F_T, F_S). \tag{4.5}$$

We pass mini-batches of source-target pairs into the model, repeat the process mentioned above, and keep alternating between different targets during training. The student decoder gradually learns from the teacher decoder and tends to be able to represent features from all target sets in the end.

### 4.3.5 Uncertainty-Guided Reconstruction

To exploit the temporal information in the input video sequences in the unlabeled target sets. and learn domain-invariant features from various targets, we follow the state-of-the-art self-supervised depth estimation work [49] and reconstruct the appearance of a target image from the viewpoint of an adjacent image by combining predicted depth, pose, and known camera intrinsic parameters. We found that naively applying this will cause inaccurate predictions, especially for fast-changing pixels in the temporal sequences. To overcome this, we propose to estimate an uncertainty map by taking the average of the reconstruction error maps of an input frame and its 4 adjacent frames and using that to further guide the reconstruction loss.

The pose regressor in our model yields the relative pose $T_{t \to t'}$ for each source view image $I_{t'}$, with respect to the target image $I_t$, from a consecutive monocular video sequence, by taking a pair of features extracted from $(I_t, I_{t'})$ as the inputs. The depth estimation

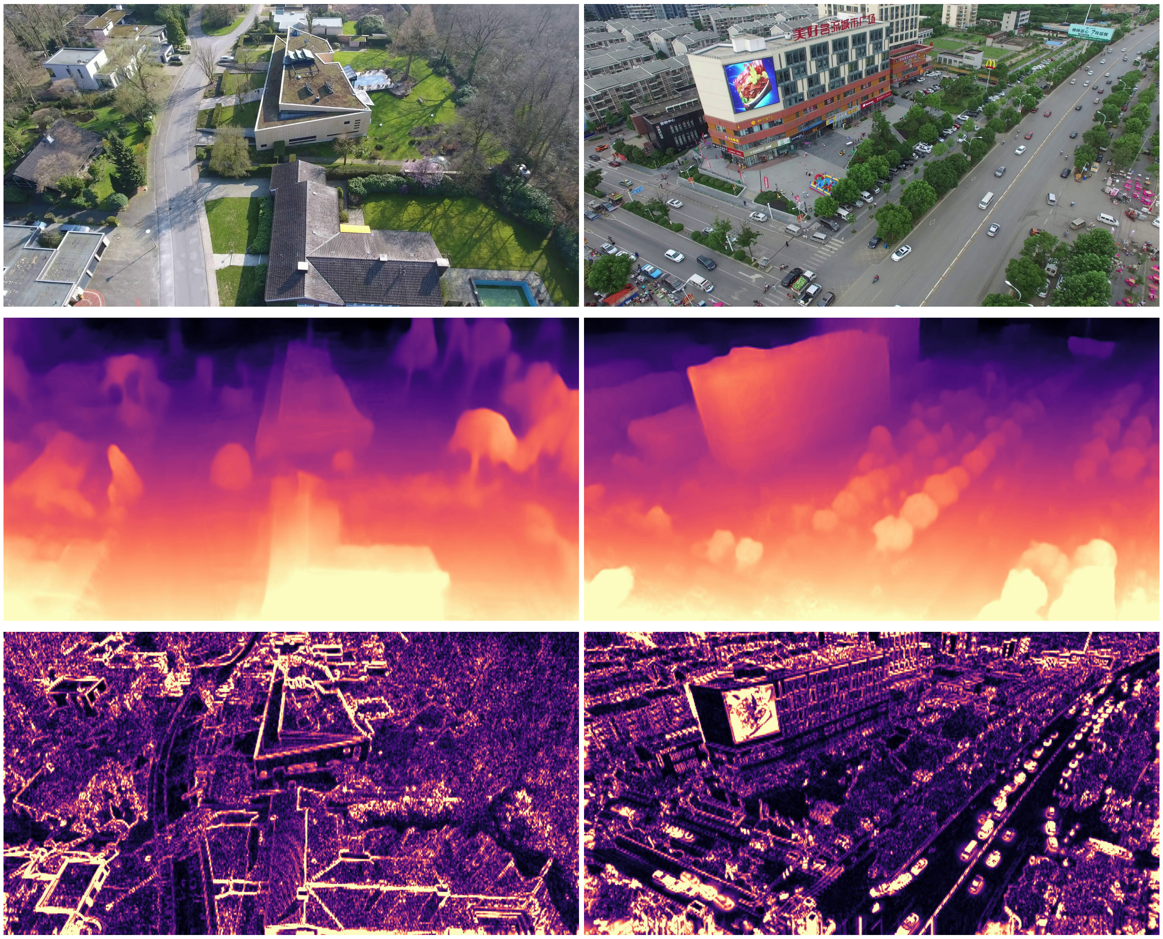


Figure 4.3: Illustration of the uncertainty maps. The 1st row shows the input images, the 2nd row shows the predicted depth maps, and the last row shows the estimated uncertainty maps, which mostly highlight rapidly-changing pixel regions including vehicles and building edges.

decoder predicts a dense depth map $D_t$ simultaneously. Our goal is to minimize the reconstruction error $L_r$, where

$$L_r \quad = \quad \sum_{t'} ||I_t - I_{t' \to t}||. \tag{4.6}$$

The image reconstruction loss, in our case, is the $\ell_1$ distance in pixel space. By using the source image $I_{t'}$, the predicted depth $D_t$, the relative pose $T_{t \to t'}$, and the camera intrinsic parameters $K$, we can reconstruct the target image $I_t$ by:

$$I_{t' \to t} \quad = \quad I_{t'} \Big\langle proj(D_t, T_{t \to t'}, K) \Big\rangle. \tag{4.7}$$

where $proj()$ are the resulting 2D coordinates of the projected depths $D_t$ in $I_{t'}$ and $\langle\rangle$ is the sampling operator.

To reduce noise in the prediction, we use edge-aware smoothness [48, 49]:

$$L_s = |\partial_x d_t^*| \, e^{-|\partial_x I_t|} + |\partial_y d_t^*| \, e^{-|\partial_y I_t|}, \tag{4.8}$$

where $d_t^* = d_t/\overline{d_t}$ is the mean-normalized inverse depth to discourage shrinking of the estimated depth. The complete self-supervised loss can be represented as:

$$\mathcal{L}'_{recon} = L_r + \lambda_s L_s \tag{4.9}$$

We propose to estimate uncertainty maps by computing the reconstruction error map generated from $I_t$ and its N (in this case N=4 is used) adjacent frames $I_{t+1}$, $I_{t+2}$, $I_{t-1}$, $I_{t-2}$. Following Eq- 4.7, the uncertainty map is estimated by taking the average of all the adjacent reconstruction error maps:

$$\mathcal{U}_t = \frac{1}{N} \sum_{i=0}^{N-1} ||I_{t_i \to t} - I_t||. \tag{4.10}$$

Note that the estimated uncertainty maps highlight drastically changing pixels (see Fig. 4.3 for illustration), e.g., edges of buildings. Therefore, those pixels with higher uncertainty are down-weighted in the reconstruction loss:

$$\mathcal{L}_{recon} = \frac{\lambda_r \mathcal{L}'_{recon}}{\mathcal{U}_t} + \mathcal{L}'_{recon}. \tag{4.11}$$

### 4.3.6 Overall Loss Function

The overall loss function of the proposed CrossAdapt framework is the weighted sum of the loss functions mentioned above and can be written as follows:

$$\mathcal{L} = \mathcal{L}_{supervised} + \alpha_1 \mathcal{L}_{adv} + \alpha_2 \mathcal{L}_{align} + \alpha_3 \mathcal{L}_{recon}. \tag{4.12}$$

## 4.4 Experiments

We evaluated our method by conducting experiments on four diverse datasets. Below we describe the datasets used for these experiments, implementation details, and extensive performance analysis.

### 4.4.1 Datasets

We describe the datasets considered in this work. While we consider the following four datasets, it is easy to extend our approach to other datasets as well.

Table 4.1: KITTI→UAVid China + UAVid Germany

| Target | Method | $\ell_1$ (prev) | $\ell_1$ (next) | SSIM (prev) | SSIM (next) |
|---|---|---|---|---|---|
| UAVid China | Monodepth2 [49] | 0.1230 | 0.1261 | 0.3181 | 0.3226 |
| | CoMoDA [70] | 0.1193 | 0.1042 | 0.2901 | 0.3009 |
| | MTKT [114] | 0.0812 | 0.0833 | 0.2216 | 0.2305 |
| | CrossAdapt (w/o reconstruction) | 0.0910 | 0.0907 | 0.2270 | 0.2299 |
| | CrossAdapt (w/o uncertainty) | 0.0629 | 0.0651 | 0.1876 | 0.1841 |
| | CrossAdapt (Ours full) | **0.0620** | **0.0513** | **0.1702** | **0.1788** |
| UAVid Germany | Monodepth2 [49] | 0.1861 | 0.1873 | 0.3909 | 0.3981 |
| | CoMoDA [70] | 0.1741 | 0.1725 | 0.3676 | 0.3755 |
| | MTKT [114] | 0.1785 | 0.1680 | 0.3601 | 0.3761 |
| | CrossAdapt (w/o reconstruction) | 0.1801 | 0.1795 | 0.3644 | 0.3606 |
| | CrossAdapt (w/o uncertainty) | 0.1581 | **0.1526** | 0.3511 | 0.3537 |
| | CrossAdapt (Ours full) | **0.1468** | 0.1541 | **0.3488** | **0.3412** |

Table 4.2: KITTI→UAVid China + UAVid Germany + Virtual KITTI

| Target | Method | $\ell_1$ (prev) | $\ell_1$ (next) | SSIM (prev) | SSIM (next) |
|---|---|---|---|---|---|
| UAVid China | Monodepth2 [49] | 0.1487 | 0.1401 | 0.3590 | 0.3574 |
| | CoMoDA [70] | 0.1386 | 0.1344 | 0.3067 | 0.3156 |
| | MTKT [114] | 0.1345 | 0.1509 | 0.2687 | 0.2459 |
| | CrossAdapt (Ours) | **0.0918** | **0.0927** | **0.2141** | **0.2108** |
| UAVid Germany | Monodepth2 [49] | 0.1762 | 0.1705 | 0.3921 | 0.3700 |
| | CoMoDA [70] | 0.1676 | **0.1609** | 0.3822 | 0.3850 |
| | MTKT [114] | 0.1887 | 0.1654 | 0.3709 | 0.3885 |
| | CrossAdapt (Ours) | **0.1531** | 0.1676 | **0.3596** | **0.3677** |
| Virtual KITTI | Monodepth2 [49] | 0.1648 | 0.1732 | 0.3390 | 0.3371 |
| | CoMoDA [70] | 0.1731 | 0.1704 | 0.3219 | 0.3232 |
| | MTKT [114] | 0.1666 | 0.1796 | 0.3395 | 0.3368 |
| | CrossAdapt (Ours) | **0.1634** | **0.1681** | **0.3183** | **0.3210** |

- **KITTI** [42]: the KITTI dataset was recorded from a vehicle while driving around Karlsruhe, Germany. In our work, we use both raw images and the corresponding depth annotations from KITTI. We follow Zhou's [151] pre-processing to remove static frames and obtained 39,810 image-depth pairs for training and 4,424 pairs for validation. Following [49], we use the same camera intrinsic parameters for all images and set the principal point to the image center and the focal length to the average of all the focal lengths in KITTI. The resolution of the KITTI raw images we use is $1242 \times 375$. In this work, we consider KITTI as the source domain.

- **Virtual KITTI** [8]: the Virtual KITTI is one of the most commonly used datasets for the task of *synthetic-to-real* domain adaptation. It contains 50 photorealistic synthetic videos, including 21,260 image-depth pairs of size $1242 \times 375$. It was created using a game engine [37] to synthesize realistic videos from the KITTI dataset. We use virtual KITTI as one of the target datasets to evaluate the performance of our model on synthetic data, and we choose to use the same camera parameters as KITTI.

- **UAVid China**[85]: UAVid is a recent aerial imagery dataset with 4K resolution. The images are captured by a drone from a low altitude. It provides ground-truth semantic labels for segmentation, in which 10 frames every 5s are labeled. UAVid contains 42 videos in total, and each video includes 900 frames. Among those videos, 31 videos were collected in China, which gives us 27,900 frames at a resolution of $3840 \times 2160$. Following [88], we use a frame rate of 1 fps to minimize noise and parallax effects.
- **UAVid Germany**[85]: UAVid Germany is also a subset of UAVid. The videos are captured in both rural and urban areas in Germany. It contains 9 aerial videos, which gives us 8,100 frames in total, at the resolution of $4096 \times 2160$. Here we use a frame rate of 10 fps to achieve optimal performance. The scene complexity of the UAVid Germany dataset is lower than UAVid China [94].

### 4.4.2 Implementation Details

We implement our model using PyTorch [99]. We follow the same publication standard of several recent MTDA works [113, 93] and report results on two adaptation scenarios: 1-source→2-target scenario and 1-source→3-target scenario. Following the existing state-of-the-art methods [49, 70], we use a similar U-Net style architecture and adopt the ResNet-18 as the feature extraction backbone to ensure a fair comparison. All networks are pre-trained on ImageNet. We also use the training protocol from previous work [49], with a learning rate of $10^{-4}$ for the first 15 epochs which is then dropped to $10^{-5}$ for the rest of the training process. For hyperparameters, the adversarial term $\lambda_{adv}$ is set to $1.0$, the smoothness term $\lambda_s$ is set to $0.001$, and the reconstruction term $\lambda_r$ is set to $0.01$. For the overall loss function, we set $\alpha_1$ and $\alpha_2$ to 0.1 and $\alpha_3$ 0.01 to maintain a balance between each term during training. For data pre-processing, we resize all input images to $640 \times 192$. For ground-levels images, including KITTI and Virtual KITTI, we apply random horizontal flipping for data augmentation. For UAVid images, we disable that due to the principal point offset to the image center [88].

### 4.4.3 Baselines

To the best of our knowledge, no directly comparable previous works have addressed the problem of multi-target domain adaptation for depth estimation. Therefore, we choose to compare with the state-of-the-art, self-supervised depth-estimation network, Monodepth2 [49], a recent single-target, domain-adaptation approach for depth estimation, CoMoDA [70], and a state-of-the-art multi-target domain adaptation for semantic segmentation approach, MTKT [114]. There are several other works on domain adaptation on depth estimation, e.g., DESC [82] and GASDA [148]. However, these works either consider using semantic labels in both source and target domains as a weak training signal, which may boost the performance or uses source-to-target image translation, which is not straightforward to perform for the multi-target setting. In our setting, we do not have any forms of labels for the target domains. The baselines we choose to compare are summarized as follows:

- **Monodepth2** [49]: Monodepth2 is considered as a solid self-supervised depth-estimation network that can generalize well to many ground-level imagery datasets. Here we use it as a naïve baseline by training it on KITTI and directly testing on our target dataset to demonstrate the significant domain gaps.

- **CoMoDA** [70]: CoMoDA is a recent state-of-the-art domain adaptation approach for depth estimation. Instead of considering the training stage adaptation only, it also considers the inference stage adaptation.

- **MTKT** [114]: MTKT is a recent state-of-the-art multi-target domain adaptation approach for semantic segmentation. We applied the proposed multi-adversarial framework to our depth estimation problem and considered that as a strong baseline.

### 4.4.4 Evaluation Metrics

Considering the fact that we do not have access to the ground-truth depth labels of UAVid for computing absolute depth errors, here we choose to report errors related to the reconstruction quality of neighboring frames in the monocular sequence, including the previous frame and the next frame, which can cross-validate how accurate the depth predictions are. More specifically, we report four metrics: $\ell_1$ (prev), $\ell_1$ (next), SSIM (prev), and SSIM (next). $\ell_1$ (prev)/(next) represents the $\ell_1$ error between the target image and the reconstructed target image from the previous/next frame. To better evaluate the structural similarity between the original and reconstructed images, we also adopt a commonly used

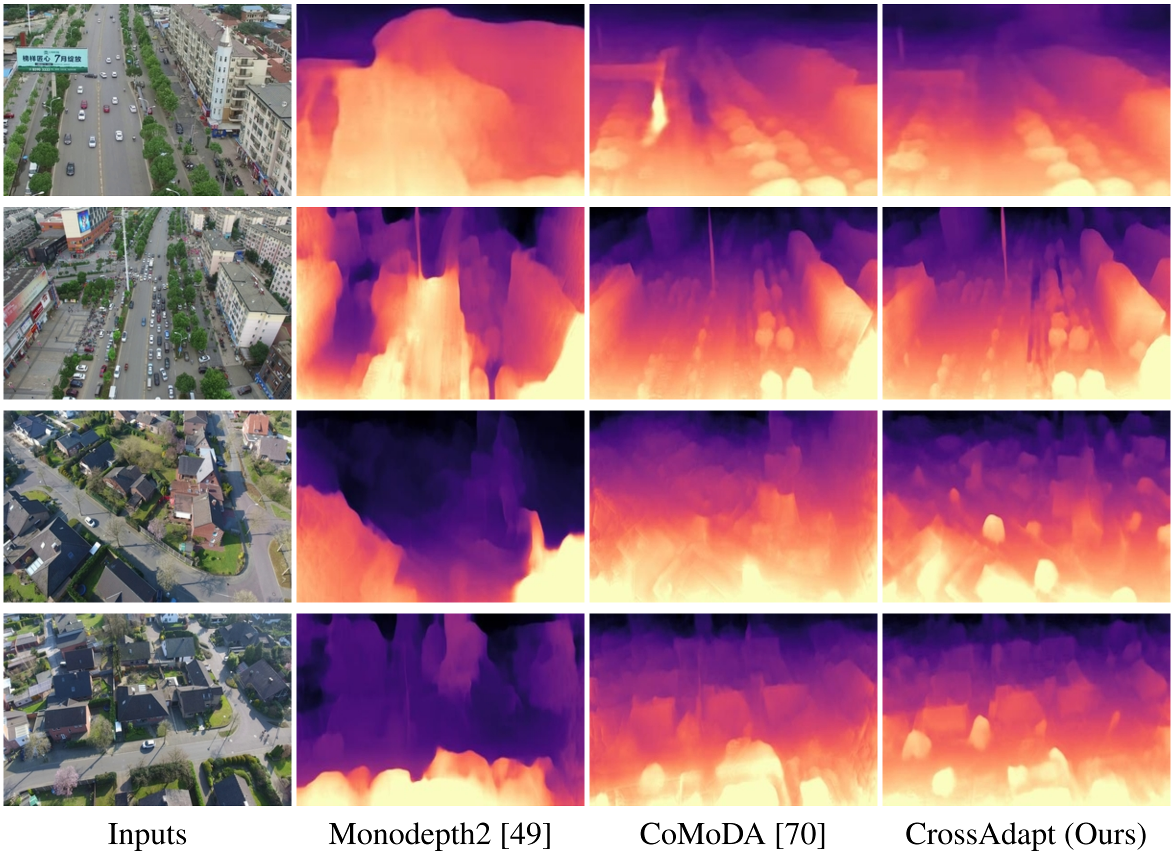


Figure 4.4: Qualitative results of KITTI→UAVid China + UAVid Germany. Both CrossAdapt and CoMoDA[70] achieve reasonable visual performance on overhead imagery, but CrossAdapt outperforms CoMoDA[70] in terms of more accurate details.

image similarity metric SSIM [59], which is shown as SSIM (prev), and SSIM (next) in the tables. Noted that the numbers we reported in the tables are 0SIM loss, which means 1-SSIM (score).

### 4.4.5 Experimental Results

We summarize the comparisons between our method and the baselines in Table 4.1 and 4.2. We use KITTI as the source for all the adaptation scenarios since it contains the most complete depth annotations. For the 1-source→2-targets scenario, we use both UAVid China and UAVid Germany as targets to evaluate the model's ability to handle extreme viewpoint changes. Both quantitative results (Table 4.1) and qualitative results (Fig. 4.4) show that our model outperforms the baselines. For the 1-source→3-targets scenario, we use UAVid China, UAVid Germany, and Virtual KITTI as targets. The results are listed in Table 4.2. We also conduct an ablation study for the first scenario (1→2), listed in Table 4.1 as well. The experiments show that the self-supervised reconstruction loss significantly improved the performance and the uncertainty guidance slightly boosted the performance when it was applied together with the reconstruction loss.

From Fig. 4.4, we can see that Monodepth2 (second column) yields crude depth estimates. This failure shows the difficulty of adaptation from ground-level to UAV imagery. CoMoDa (third column) performs better and captures rough outlines of buildings, but it often merges adjacent buildings. Predicted depth maps from our method, CrossAdapt (fourth column) are able to accurately separate buildings and get fine estimates for trees.

## 4.5 Conclusion

We introduced CrossAdapt, a novel approach to multi-target domain adaptation for the task of monocular depth estimation. A key feature of our approach is the use of a combination of the student-teacher model and uncertainty-guided self-supervised reconstruction, which enables training on video sequences without ground-truth depth. These, together with the supervised depth estimation for the source, result in a strong performance on unlabeled target domain imagery.

We evaluated our method on four diverse datasets, including two from the ground-level perspective and two from an aerial perspective. This extreme viewpoint shift is important to address given the need for UAVs to operate at many altitudes and for teams of UAVs and UGVs. Our approach provides a strong foundation for the creation of general-purpose image understanding systems that can operate across many viewpoints. In future work, we plan to explore the multitask setting, including semantic segmentation and object detection, which we expect to yield improvements in performance metrics at the expense of additional computational cost and model complexity.

# Chapter 5

# CrossSeg: Cross-Scene Few-Shot Aerial Segmentation Using Probabilistic Prototypes

In this work, we propose a novel framework called CrossSeg that addresses the task of few-shot semantic segmentation for different aerial imagery. Conventional semantic segmentation approaches struggle to generalize well to unseen object categories, making them a significant limitation for modern intelligent systems, especially those deployed in realistic real-time settings, such as unmanned aerial vehicles (UAVs). CrossSeg overcomes this limitation and generalizes well in a cross-scene setting with only a few labeled samples. Unlike traditional methods that use a set of fixed prototypes for each class, CrossSeg utilizes high-quality probabilistic prototypes that can not only represent different semantic classes but also handle significant variations in different scenes. We evaluate our method on four datasets, Potsdam, Vaihingen, Aeroscapes, and UAVid, which contain widely varying viewpoints and are more challenging than those considered in prior work on few-shot learning for semantic segmentation. Our approach significantly improves upon conventional few-shot segmentation baselines and does not require extensive tuning.

## 5.1 Introduction

Deep neural networks have achieved impressive performance on a variety of vision-related tasks, including image classification, object detection, and semantic segmentation. Among those tasks, semantic segmentation is usually considered the most challenging one, mostly because of the requirement of categorizing every single pixel. In recent years, people have proposed different types of methods to tackle tasks around semantic segmentation [80] [10] [138], and these methods are primarily designed to recognize the classes that have shown up in the training phase, not unseen classes. However, modern intelligent autonomous systems, especially unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), often require the models deployed on devices should be able to adapt quickly to the surrounding environment and make reasonable predictions even for unseen objects and classes without extensive human intervention or tuning. This is particularly important for aerial applications—images collected by drones and satellites are often under drastically different conditions, e.g., scenes, weather, light, camera poses, and geographic location, and it is often impossible to manually label all pixels for those data to train a model. Failing in recognizing unseen objects may lead to inaccurate behaviors or even severe safety issues, especially for fast-moving UAVs. To tackle this problem, we propose to build a model that can quickly learn from a very limited number of annotated images to predict reasonable pixel-level semantic labels across different scenes. More specifically, we aim to provide a few-shot learning framework for semantic segmentation that can be applied to multiple aerial and remote sensing scenes.

Semantic segmentation, which is also referred to as image classification in the remote sensing field, aims to assign pixel-level labels to the input images. Typically, training such segmentation models requires massive datasets with pixel-level labels, which are usually expensive to acquire. To deal with this challenge, weakly supervised, semi-supervised, and unsupervised methods are proposed but these are still difficult to generalize well to unseen categories without extensive tweaking and tuning. Instead of learning from massive annotated data, few-shot learning methods are designed to identify objects from novel categories by only looking at a few labeled samples, as humans can do. And few-shot learning methods can be useful for many aerial scenarios, for example, it is easy to find a segmentation model trained for a certain class, for instance, a road extractor, but people may aim to perform segmentation on other categories, e.g., buildings, without needing a large amount of training data. To solve such problems, few-shot learning models are not trained to remember any certain class, but to know the difference between objects from different classes.

Existing few-shot semantic segmentation methods [129] [62] usually extract feature vectors from the support images and compute a fixed set of prototypes from those vectors before applying a distancing function to segment images from the query set. However, such methods have several limitations and do not always generalize well, especially to complex aerial images. First, a deterministic class prototype can not always be the most representative feature of pixels from that class, especially when the prototype is learned from a small set. Second, during the inference stage, using the averaged features as class prototypes does not consider the confidence of every prediction made by the model. To tackle these two limitations, we propose a probabilistic prototype-based framework for few-shot semantic segmentation. Our method models the class prototypes as a probabilistic distribution instead of a set of fixed vectors, which is more robust to the uncertainty brought by the limited number of labeled support samples. Specifically, a prior net is used to learn class-specific probabilistic prototypes from the support set, and a true posterior distribution of the query set is further learned by a posterior net to guide the prior net. The prototypes sampled from the prior distribution are used for predicting query images. Furthermore, instead of using the groundtruth labels of the support set only for masking, we propose to use a cycle-regularization module to reverse the query and support sets by sampling prototypes from the posterior distribution. During inference, unlike the previous methods which usually use the averaged feature from each class as the prototype, we adopt a confidence-weighted strategy to predict the final segmentation map. We evaluate our method on multiple diverse aerial datasets, including Potsdam, Vaihingen, Aeroscapes, and UAVid, and results show that our method achieves significant improvement over the state-of-the-art cross-domain and in-domain settings.

We list the main contributions of the proposed method as follows.

- We introduce a novel probabilistic prototype-based few-shot semantic segmentation framework for aerial applications.
- Our method proposes to use a cycle-regularization module to further exploit the annotations of the support set.
- We propose to use a confidence-weighted strategy to predict segmentation maps during inference.
- We explore four diverse aerial and remote sensing datasets and conduct experiments on both in-domain and cross-domain settings, and demonstrate the robustness of our model.

## 5.2 Related Works

In this section, we introduce three topics relevant closely to our work, including the origin and development of few-shot learning, important semantic segmentation models proposed in recent years, and how the existing few-shot semantic segmentation models relate to our proposed method.

### 5.2.1 Few-shot Learning

Few-shot learning aims to learn universal knowledge that is easy to transfer to new domains and classes with only a few annotated samples. Most of the recent works are based on deep neural networks and primarily include two types: metric-learning-based methods [117] [126] and optimization-learning-based methods [105] [32]. Metric-learning-based methods aim to encode input samples into an embedded feature space and to perform the distancing matching for classification [126]. The prototypical network [117] further improves this idea by learning a metric space where the input is classified based on its distance to the prototypical features of each class. Optimization-learning-based type of methods, which are also referred to as meta-learning-based methods, consider the inconsistency between the training and test set and tackle this problem by tuning the model quickly in the test phase [105] [32]. Our work follows the few-shot learning setting and extends it to the task of semantic segmentation.

### 5.2.2 Semantic Segmentation

Semantic segmentation aims to assign a set of predefined semantic classes to every pixel in the input images. CNNs-based methods [80] [75] [3] [147] [11] achieve great success in this field. For example, FCN [80] first adopts deep CNNs and proposes a fully convolutional network which improves segmentation performance by a large margin. Dilated convolutions [138] [11] are widely applied to increase the size of the receptive field without losing spatial information. Another interesting line of work is based on the conditional variational autoencoder (C-VAE) [118]. C-VAE extends VAE [66] into the conditional generative model for supervised learning.

More recently, probabilistic models have been [141] [33] [149] introduced to CNN-based models to handle the uncertainty caused by scarce training data, and C-VAE has also been broadly applied in segmentation tasks. For example, the probabilistic U-net [67] which combines U-Net [108] with C-VAE for semantic segmentation achieves great success in handling the ambiguities existing in medical images. An extension [68] of the probabilistic U-Net introduces a hierarchical graphical model to decompose the latent space. A similar framework [89] proves that this strategy can also be applied to instance segmentation. In this work, we follow CNN-based segmentation networks, adopt the idea of C-VAE to the few-shot learning scenario, and propose a novel few-shot segmentation framework to generalize the powerful segmentation networks to new domains and classes with only a few annotated samples.

### 5.2.3 Few-shot Semantic Segmentation

Few-shot segmentation is a more demanding yet challenging task when compared to few-shot classification. The initial few-shot segmentation models [115] [104] usually generate a group of parameters from the support images using a conditioning branch, merge the extracted support features with the query features, and then use a fine-tuned decoder to predict segmentation maps. Other approaches [143] [60], utilize either a masked average pooling strategy or an attention module to extract background and foreground features from the support set more effectively. In general, these methods adopt a parametric module, which combines information extracted from the support set to generate segmentation maps.

A similar approach [25] was proposed to solve the problem of few-shot segmentation using a variant of the prototypical network. This approach uses the concept of prototypical learning and metric learning to solve the problem, but it has a complicated configuration and a three-stage training pattern. Additionally, their method focuses on extracting prototypes based on image-level losses and using them as guidance for fine-tuning the segmentation of the query set. In contrast, our model adopts a simplified end-to-end design and is more closely aligned with the original Prototypical Network [117]. Furthermore, we propose a probabilistic method for generating class prototypes, which is modeled as a distribution instead of a single deterministic vector, making the model more robust to adapt to cases with sparse or changing annotations. We further introduce a cycle-regularization scheme, which is learned for exploiting the annotations from both support and query sets. We will introduce our approach in detail in the next section.

## 5.3 Approach

In this section, we introduce our method CrossSeg, a few-shot semantic segmentation model using probabilistic prototypes. We explain the design of each component and the training and inference process in detail. The illustration of our method can be found in Fig. 5.1.

### 5.3.1 Problem Statement

We aim to propose a method for quickly adapting a segmentation model to perform segmentation on unseen objects using only a few annotated samples. We follow recent approaches [62] [129] in the field and adopt a standard training and inference protocol. The images are divided into two non-overlapping sets of classes: $\mathcal{C}_{\text{seen}}$ and $\mathcal{C}_{\text{unseen}}$. The training set $\mathcal{D}_{\text{train}}$ is constructed from $\mathcal{C}_{\text{seen}}$ and the test set $\mathcal{D}_{\text{test}}$ is constructed from $\mathcal{C}_{\text{unseen}}$. The segmentation model is trained on $\mathcal{D}_{\text{train}}$ and evaluated on the unseen set $\mathcal{D}_{\text{test}}$. Both the training set $\mathcal{D}_{\text{train}}$ and test set $\mathcal{D}_{\text{test}}$ contain numerous *episodes*, each of which consists of a set of annotated support images $\mathcal{S}$ and a set of query images $\mathcal{Q}$. Specifically, $\mathcal{D}_{\text{train}} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{test}} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{N_{\text{test}}}$, where $N_{\text{train}}$ and $N_{\text{test}}$ represent the number of episodes for training and test respectively.

We formulate our task as a $\mathcal{C}$-way $\mathcal{K}$-shot learning segmentation task. Specifically, $\mathcal{C}$-way represents that there are $\mathcal{C}$ different classes in total, and $\mathcal{K}$-shot means that there are $\mathcal{K}$

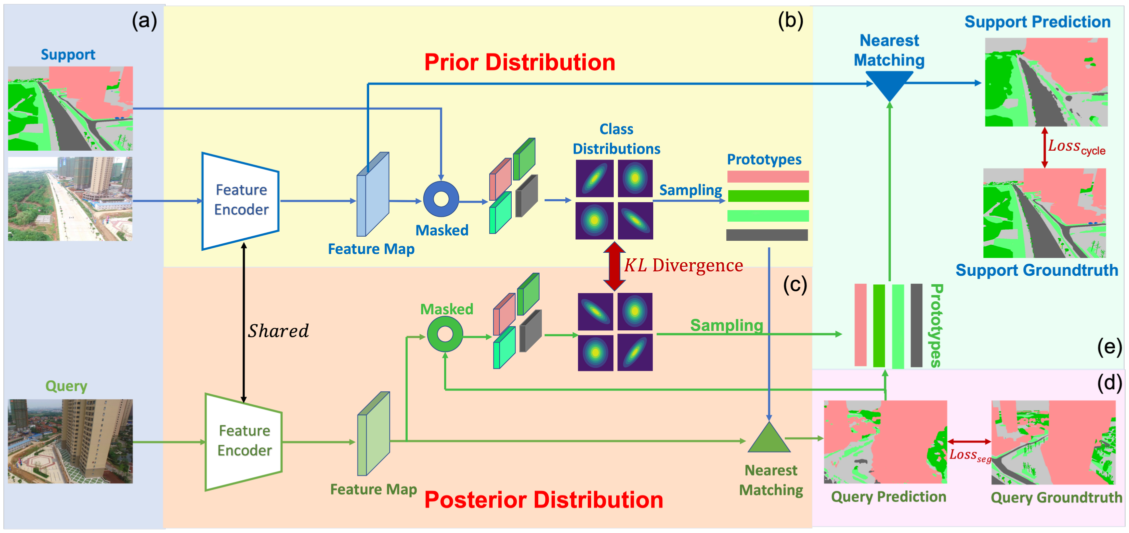


Figure 5.1: Overview of our method. (a) The support images and the corresponding class masks are passed to the prior network. The query images are passed into the posterior branch. (b) A prior encoder takes as input the support image-label pairs and generates semantic features and the probabilistic distribution of class-specific prototypes. The support masks are used to guide the prototype pooling. (c) The posterior encoder takes as input the query images and predicts the true distribution of query prototypes and semantic features for further segmentation. (d) The groundtruth labels of query images are used to compute the supervised semantic loss. (e) To exploit the support information, we use a cycle-alignment module to use query predictions as pseudo masks to generate support prototypes. Those prototypes are then used to segment the semantic features from the support set.

images per semantic class in each episode. A support set consists of $N_{\text{support}}$ image-mask pairs and can be represented as $\mathcal{S}_i = \{(I_{c,k}, M_{c,k})\}$, where $k = 1, 2, \cdots, K$ and $c \in \mathcal{C}_i$ with $|\mathcal{C}_i| = C$. The query set $\mathcal{Q}_i$ consists of $N_{\text{query}}$ image-mask pairs from the same set of classes in the support set. We train our model to learn knowledge from the support set and then use that knowledge to segment images in the query set. Noted that each episode consists of different semantic classes, so the model is not trained to learn those class-specific features but to learn the differences between classes and generalize well new classes. In general, our goal is to train the segmentation model on the training set $\mathcal{D}_{\text{train}}$, and evaluate its performance on the test set $\mathcal{D}_{\text{test}}$ by only seeing a few annotated samples from the novel classes. Specifically, during evaluation, the model is further tuned on the labeled support set $\mathcal{S}_i$ from each test episode in $\mathcal{D}_{\text{test}}$ and evaluated on the unseen query set $\mathcal{Q}_i$.

### 5.3.2 Method Overview

In the context of few-shot learning, our model is geared to handle new classes that were not encountered during model training. Instead of learning a set of class-specific representations, the model should have the capability to acquire transferable knowledge for seen and unseen classes.

To achieve this goal, our model is designed to learn from support-query pairs and it models class prototypes as probabilistic distributions rather than a set of fixed vectors, making it more robust to the uncertainty introduced by the limited number of labeled support samples. As shown in Fig. 5.1, we first pass the support images and the corresponding groundtruth masks into the prior net, followed by a semantic decoder to generate support predictions and a prior generator ($G_{prior}$ in Fig. 5.1) to embed the support set into a latent space. The class prototypes are sampled from this space and then used together with the query features to generate query semantic segmentation map predictions. In general, the proposed method comprises two key components: a prior network that learns class-specific probabilistic prototypes from the support set and a posterior network that infers the true posterior distribution of the query set. The two networks are trained together in an end-to-end fashion, where the prototypes generated by the prior network are used for predicting the segmentation of query images, and the guidance from the posterior network helps tune the prior network. The network is also supervised by the semantic loss computed between the query prediction and the query groundtruth annotations. Additionally, instead of utilizing the groundtruth labels of the support set solely for masking, we propose to use a cycle-alignment module (illustrated in Fig. 5.1 (e)) to reverse the roles of the query and support sets by sampling prototypes from the posterior distribution, which are then utilized to predict support segmentation maps. This scheme helps further exploit the support information.

### 5.3.3 Prior-Posterior Architecture

The key component of our model is the prior-posterior architecture, which includes a prior network estimating the prototype distribution for each class and a posterior network generating the true distributions of query images. Specifically, the prior net deploys a CNN encoder to extract representations of support set images, which are then projected into a latent space by a generator. Here we make an assumption that the prototype latent space for each class follows a multivariate Gaussian distribution with a mean and a diagonal covariance structure. The groundtruth annotations over the support images are used to pool the prototype for each category separately. We follow PANet [129] and adopt a late fusion strategy, which masks the feature maps to produce class-wise features separately instead of directly masking the images before feeding them into the networks. We pool the feature map for each category and get a fix-sized vector $c$ as the class representation. The generator directly inputs $c$ and splits the output into two equal dimensional vectors, $\mu$ and $\sigma^2$. To further constrain the range of the variance vector $\sigma^2$, we propose to rescale it by using a Sigmoid function $S$ with learnable parameters $w$ and $b$. Therefore, the learned latent space parameters can be represented as:

$$[\mu_{prior}, \sigma^2] = G_{prior}(c), \tag{5.1}$$

$$\sigma^2_{prior} = w \cdot S(\sigma^2) + b. \tag{5.2}$$

The idea behind the prior network is to sample multiple prototypes as we need, which increases the generalization capability of the class-specific prototypes. However, directly guiding the query prediction by using such distributions generated from the support set without supervision is not always reliable.

Therefore, we propose to use a posterior net to generate true prototype distributions of the query set. Similar to the prior net, the posterior net uses the same CNN encoder to extract the features of query images and then uses predicted segmentation masks as pseudo masks to obtain the true prototype distribution for each class. Finally, a mean vector $\mu_{post}$ and a variance vector $\sigma^2_{post}$ are output from the posterior net for the posterior distribution using the same protocol as the prior net.

We minimize the KL divergence to bridge the gap between the prior and the posterior distribution. This encourages the inferred prototypes from the support images to match those from the query images.

$$\mathcal{L}_{KL} = D_{\mathrm{KL}}[q(\mathbf{z}_q|Q)||p(\mathbf{z}_p|S)], \tag{5.3}$$

where $q$ and $p$ represent the posterior (query) and prior (support) net respectively, and $z_q$ and $z_p$ represent the predicted class prototypes.

### 5.3.4 Cycle-Alignment

Instead of using the groundtruth annotations from the support set only for masking [115] [104] [128], we propose a cycle-alignment module to further exploit those annotations by reversing the query and support images and sampling prototypes from the posterior distributions during training. In previous works, the support annotations are used only for masking, which actually does not adequately exploit the support information for few-shot learning. This design is intended to exploit the information from the support set and improve the ability of the model to generalize from limited examples.

It makes sense that if the model is able to effectively predict a segmentation mask for the query image using prototypes learned from the support set, then the prototypes derived from the query set based on the predicted masks should be capable of producing good segmentations on the support images. Therefore, our cycle-alignment module makes the segmentation network perform few-shot learning in a reverse direction, that is, treating the query and its predicted mask as the new support set to segment the original support images. It is important to note that this process only happens during training, and all the support and query images used are taken from the training set $\mathcal{D}_{\text{train}}$.

The cycle-alignment module is illustrated in Fig. 5.1 (e). Once a predicted segmentation map is made for the query image, we use it to pool the query features and sample prototypes from the posterior latent space. This results in a new set of prototypes. Subsequently, these prototypes are used to predict the segmentation masks for the support images in a non-parametric way. Finally, the cycle-alignment loss $\mathcal{L}_{\text{cycle}}$ is calculated by comparing the support predictions with the support groundtruth annotations. This cycle-alignment approach is essentially swapping the roles of the support and query sets, and by doing that the model is encouraged to learn a uniform latent space that aligns support and query prototypes.

### 5.3.5 Training and Inference

During training, we iterate every episode that contains support and query sets. We sample class-specific prototypes from the prior distribution, use them to predict the segmentation maps of query images, and compute the supervised segmentation loss $\mathcal{L}_{seg}$. The posterior network utilizes the predicted masks to generate the posterior distribution, and the KL divergence is used to minimize the distance between the prior and posterior distributions. Finally, the predicted segmentation maps of the query images are used to generate query prototypes, which are subsequently employed for the cycle-alignment procedure outlined above. The total loss for training our model is thus

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_0 \mathcal{L}_{\text{KL}} + \lambda_1 \mathcal{L}_{\text{cycle}} \tag{5.4}$$

where $\lambda_0$ and $\lambda_1$ serve as loss weights. In our experiments, we keep both weights as $1$ since different values give little improvement.

During the inference phase, we have a few labeled support images from $\mathcal{D}_{\text{test}}$ and our goal is to segment the unlabeled query images from the same set. Specifically, we need to obtain a set of high-quality class prototypes from the limited number of labeled samples and use them to predict segmentation maps of query images. For the $k$-shot setting, we generate a prior distribution $\{\mathcal{N}_i(\mu_i, \sigma_i^2)\}_{i=1}^{k}$for each of the $k$ support image-mask pairs.

Subsequently, we adopt a variance-weighted average strategy and generate an overall distribution:

$$\mu = \frac{\sum_{i=1}^{k} \frac{1}{\sigma_i^2} \mu_i}{\sum_{i=1}^{k} \frac{1}{\sigma_i^2}}, \tag{5.5}$$

$$\sigma^2 = \frac{k}{\sum_{i=1}^{k} \frac{1}{\sigma_i^2}}. \tag{5.6}$$

The variance-weighted average operation differs from the equal-weighted average operation by assigning greater weight to distributions with lower variance. This leads to the enhancement of more representative distributions while limiting the impact of less significant ones.

## 5.4 Evaluation

In this section, we present an overview of the datasets employed in our research, highlight the baseline methods we used for comparison, and display the results and analysis of our experiments.

### 5.4.1 Dataset

- **Potsdam&Vaihingen** [109]: This dataset is usually referred to as ISPRS, and it contains very high-resolution aerial images collected in two cities of Germany: Vaihingen and Potsdam. And for each aerial image, the ground truth labels are provided on six classes: buildings, impervious surfaces (Imp. surf.), low vegetation (Low veg.), trees, cars, and clutter.

- **Aeroscapes** [95]: Aeroscapes is a large labeled aerial dataset for semantic segmentation. There are 3269 labeled images with 12 classes. Image size is $1280 \times 720$. Most images are captured in an urban setting.

- **UAVid**[86]: UAVid is a recent aerial imagery dataset with 4K resolution. The images are captured by a drone from a low altitude. It provides ground-truth semantic labels for segmentation, in which 10 frames every 5s are labeled. UAVid contains 42 videos in total, and each video includes 900 frames. Among those videos, 31 videos were collected in China, which gives us 27,900 frames at a resolution of $3840 \times 2160$. Following [88], we use a frame rate of 1 fps to minimize noise and parallax effects. UAVid Germany videos are captured in both rural and urban areas. It contains 9 aerial videos, which gives us 8,100 frames in total, at the resolution of $4096 \times 2160$. Here

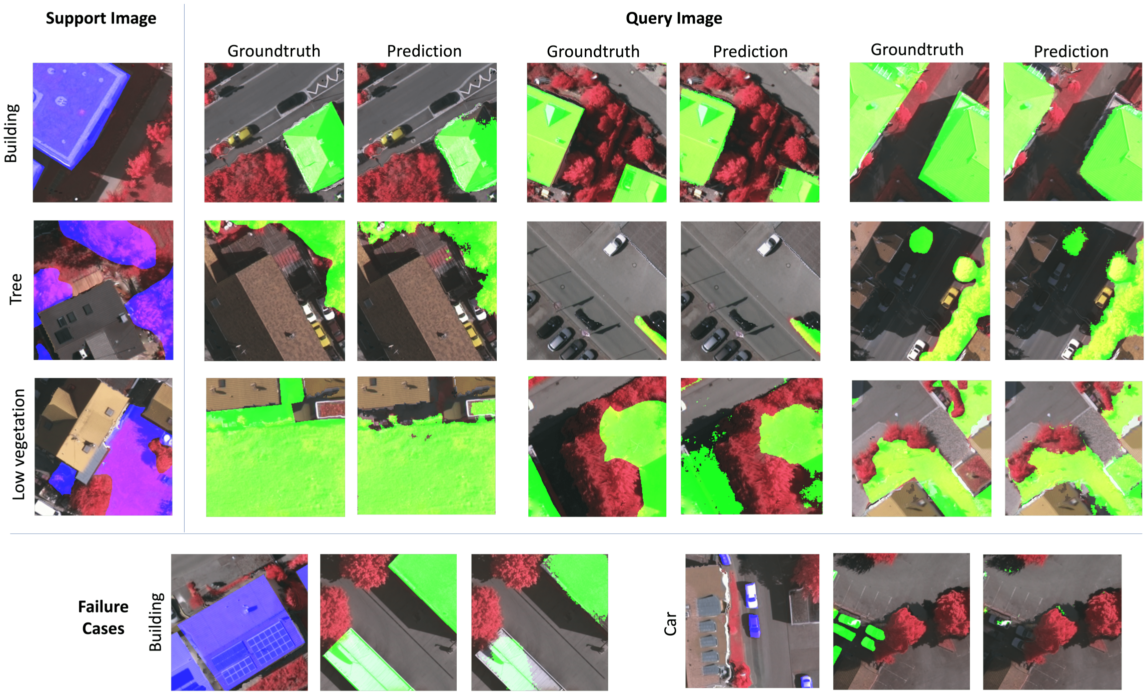


Figure 5.2: Qualitative results of 1-way 5-shot learning for cross-domain on Vaihingen dataset. The first three rows represent building, tree, and low vegetation respectively. And the bottom row represents two failure cases, caused by the strong light reflection or the shadow. Class annotations are blue for support images and green for query images.

we use a frame rate of 10 fps to achieve optimal performance. The scene complexity of the UAVid Germany dataset is lower than UAVid China [94].

### 5.4.2 Implementation Details

We deploy the ResNet50 backbone pre-trained on ImageNet as the encoder. The prior encoder and the posterior encoder share weights during training. The generator is implemented by using two fully connected layers, for predicting mean and variance respectively. Input images are resized to (417, 417) and augmented using random horizontal flipping. The model is trained end-to-end by SGD with a momentum of 0.9 for 30,000 iterations. The learning rate is initialized to 1e-3 and reduced by 0.1 every 10,000 iterations. The weight decay is 0.0005 and the batch size is 1. Noted that we consider the mask of each support image to be a binary mask (background and foreground). We compare with two state-of-the-art few-shot semantic segmentation methods, PANet [129] and FPS [62], as baselines. We report the F1 score of each class to evaluate the segmentation performance.

Table 5.1: Cross-Domain Evaluation: Results on the Vaihingen Datasets.

| Methods | Imp. Surf | Buildings | Low Veg. | Trees | Cars | Overall |
|---|---|---|---|---|---|---|
| FCN [80] (oracle) | 90.5 | 93.7 | 83.4 | 89.2 | 72.6 | 89.1 |
| PANet [129] | 51.4 | 71.1 | 39.6 | 75.7 | 23.5 | 62.1 |
| FPS [62] | 62.6 | 73.1 | 38.7 | 80.5 | 41.1 | 67.7 |
| Ours | **64.5** | **75.8** | **39.1** | **82.1** | **43.0** | **70.1** |

Table 5.2: Cross-Domain Evaluation: Results on the Potsdam Datasets.

| Methods | Imp. Surf | Buildings | Low Veg. | Trees | Cars | Overall |
|---|---|---|---|---|---|---|
| FCN [80] (oracle) | 92.5 | 96.4 | 86.7 | 88.9 | 94.7 | 90.3 |
| PANet [129] | 49.7 | 66.7 | 40.9 | 60.1 | 20.0 | 52.5 |
| FPS [62] | 50.6 | 67.7 | 40.5 | 60.3 | 24.9 | 53.2 |
| Ours | **52.4** | **69.3** | **40.1** | **62.6** | **27.0** | **54.8** |

Table 5.3: Cross-Domain Evaluation: Results on the Aeroscapes Datasets.

| Methods | Car | Road | Construction | Vegetation | Road | Overall |
|---|---|---|---|---|---|---|
| PANet [129] | 52.3 | 63.1 | 38.7 | 65.6 | 54.1 | 53.2 |
| FPS [62] | 53.8 | 68.4 | 40.7 | 64.7 | 55.9 | 55.6 |
| Ours | **55.1** | **70.4** | **43.1** | **66.9** | **57.8** | **59.8** |

### 5.4.3 Cross-Domain Evaluation

We evaluate the cross-domain performance of our model by training the model on PASCAL VOC 2012 [30] while testing on multiple aerial datasets in a few-shot learning setting. We follow the experimental settings used in [129] and use 15 classes from PASCAL-5i during training. Due to the extremely large sizes of aerial images, we follow [62] and split the images into smaller (417 × 417) patches. We adopt the standard episodical learning scheme and conduct 5-shot learning experiments. We list the performance of our proposed method of Vaihingen in Table 5.1 and Potsdam in Table 5.2. The FCN [80] (oracle) was trained using the full training set in a fully supervised way, and we consider that as the upper bound of our method. We also conduct experiments on Aeroscapes [95] and UAVid [86]. Performances show that our method surpasses the previous methods FPS [62] and PANet [129] by a large margin on all datasets. We also include a qualitative evaluation of the Vaihingen

Table 5.4: Cross-Domain Evaluation: Results on the UAVid Datasets.

| Methods | Building | Tree | Low Veg. | Human | Road | Overall |
|---|---|---|---|---|---|---|
| PANet [129] | 54.7 | 60.4 | 55.3 | 47.5 | 69.3 | 58.5 |
| FPS [62] | 58.1 | 63.6 | 56.8 | 46.4 | 70.6 | 61.7 |
| Ours | **60.7** | **66.5** | **57.9** | **49.9** | **73.3** | **65.4** |

Table 5.5: Cross-Domain vs. In-Domain Segmentation performance on Building and Imp. Surface Classes of Vaihingen Dataset.

| Method | Category | Cross-Domain | In-Domain |
|---|---|---|---|
| FPS | building | 73.1 | 77.4 |
| | imp. surface | 62.6 | 72.0 |
| Ours | building | **75.8** | **78.3** |
| | imp.surface | **64.5** | **73.1** |

Table 5.6: Assessing the impact of changing the dimensionality $N$ of the sampled prototypes.

| Dimension | N=3 | N=6 | N=12 |
|---|---|---|---|
| Performance | 76.9 | 78.3 | 78.1 |

dataset, both successful and failure cases are illustrated in Fig. 5.2.

### 5.4.4 In-Domain Evaluation

Beyond the cross-domain evaluation, we also test our model under the in-domain setting. Following FPS [62], we train the model on selected classes from Vaihingen and evaluate the segmentation performance on other classes from the same dataset with 5 annotated images per class. We conduct experiments with two settings: 1) we train our model on the classes including impervious surface, low vegetation, tree, and car and test the performance on building class and 2) we train our model on the classes including building, low vegetation, tree, and car and test the performance on impervious surface. Table 5.5 lists the performance of these two experiments conducted on Vaihingen. The results show that our model performs better than the baseline learning with only 5 annotated labels.

### 5.4.5 Prototype Dimension

We sample prototypes from the latent space with the dimension of $N$, and then broadcast the prototypes to the same size as the semantic feature maps. We analyze the size of the latent vector $N$ and conduct experiments under the in-domain evaluation protocol using the building class in the Vaihingen dataset. The results are shown in Table 5.6. We see that even as the latent vector size increases, the performance of our method remains roughly stable.

Table 5.7: Ablation Study: 5-shot cross-Domain Evaluation on the Vaihingen Dataset.

| Method | Buildings |
|---|---|
| PANet [129] | 71.1 |
| FPS [62] | 73.1 |
| Ours (w/o prior-posterior) | 73.3 |
| Ours (w/o cycle-alignment) | 74.9 |
| Ours (full) | 75.8 |

### 5.4.6 Ablation Study

To verify the effectiveness of the two major components in our model, we conduct an ablation study listed in Table 5.7. We perform 1-way 5-shot learning on the Vaihingen dataset under the cross-domain setting, and the results show that both the prior-posterior and the cycle-alignment components bring improvement, although the prior-posterior part brings more.

## 5.5 Conclusion

We introduce a novel cross-scene few-shot semantic segmentation framework for aerial imagery. Our method can perform segmentation for unseen object categories with only a few annotated samples. This is important for autonomous systems, especially for those deployed in a realistic real-time setting, e.g., unmanned aerial vehicles (UAVs). We present CrossSeg: a novel framework that learns a semantic segmentation network that can generalize well in a cross-scene setting with only a few labeled samples. Instead of using a set of deterministic prototypes, CrossSeg offers high-quality probabilistic prototypes which can not only represent different semantic classes but can also enhance the huge variations in aerial images. We provide both quantitative and qualitative evaluations on multiple aerial and remote sensing datasets. These datasets contain widely varying viewpoints, which is more challenging than is typically considered in prior work on few-shot learning for semantic segmentation. Our approach significantly improves upon conventional few-shot segmentation baselines and does not require extensive tuning. We believe this work will be useful for many aerial and remote sensing applications.

# Chapter 6

# Discussion

Building a robust deep-learning framework for visual tasks is a challenging problem. While methods for unsupervised domain adaptation have had great success, semi-supervised domain adaptation and multi-domain adaptation without further refinement steps have yet to catch up. Our dissertation focused on improvements to those problems and settings. We proposed a novel method for semi-supervised domain adaptation, multi-domain depth estimation, and multi-domain few-shot aerial segmentation.

## 6.1 Findings

In Chapter 2, we introduce the technical background used in this dissertation, specifically convolutional neural networks, domain adaptation, self-supervised reconstruction, and prototypical learning. We also introduce a clinical case for unsupervised domain adaptation on mammogram imaging in Section 2.2.1, which is one of my early works in the Ph.D. study.

In Chapter 3, we introduce a semi-supervised domain adaptation framework for image classification. Most research on domain adaptation has focused on the purely unsupervised setting, where no labeled examples in the target domain are available. However, in many real-world scenarios, a small amount of labeled target data is available and can be used to improve adaptation. We address this semi-supervised setting and propose to use dynamic feature alignment to address both inter- and intra-domain discrepancy. Unlike previous approaches, which attempt to align source and target features within a mini-batch, we propose to align the target features to a set of dynamically updated class prototypes, which we use both for minimizing divergence and pseudo-labeling. By updating based on class prototypes, we avoid problems that arise in previous approaches due to class imbalances. Our approach, which doesn't require extensive tuning or adversarial training, significantly improves the state of the art for semi-supervised domain adaptation. We provide a quantitative evaluation on two standard datasets, DomainNet and Office-Home, and performance analysis.

In Chapter 4, we introduce a cross-scene depth estimation framework under the multi-domain setting. We address the task of monocular depth estimation in the multi-domain setting. Given a large dataset (source) with ground-truth depth maps, and a set of unlabeled datasets (targets), our goal is to create a model that works well on unlabeled target datasets across different scenes. This is a challenging problem when there is a significant domain shift, often resulting in poor performance on the target datasets. We propose to address this task with a unified approach that includes adversarial knowledge distillation and uncertainty-guided self-supervised reconstruction. We provide both quantitative and qualitative evaluations on four datasets: KITTI, Virtual KITTI, UAVid China, and UAVid Germany. These datasets contain widely varying viewpoints, including ground-level and overhead perspectives, which is more challenging than is typically considered in prior work on domain adaptation for single-image depth. Our approach significantly improves upon conventional domain adaptation baselines and does not require additional memory as the number of target sets increases.

In Chapter 5, we introduce a cross-scene few-shot semantic segmentation framework for aerial images. Conventional semantic segmentation approaches can only recognize the classes at test time that have appeared in the training set and are hard to generalize well to unseen object categories. This is a significant limitation for autonomous systems, especially for those deployed in a realistic real-time setting, e.g., unmanned aerial vehicles (UAVs). In this work, we address the task of few-shot semantic segmentation for different aerial scenes. We present CrossSeg: a novel framework that learns a semantic segmentation network that can generalize well in a cross-scene setting with only a few labeled samples. Instead of using a set of deterministic prototypes, CrossSeg offers high-quality probabilistic prototypes which can not only represent different semantic classes but can also enhance the huge variations in aerial images. We provide both quantitative and qualitative evaluations on multiple aerial and remote sensing datasets. These datasets contain widely varying viewpoints, which is more challenging than is typically considered in prior work on few-shot learning for semantic segmentation. Our approach significantly improves upon conventional few-shot segmentation baselines and does not require extensive tuning.

## 6.2 Future Works

This dissertation proposed several methods for domain adaptation and few-shot segmentation. Our work focused on the multi-domain setting, primarily using prototypes. There are several possible future research directions for extending this work. One line of research that has become popular is using Graph Neural Network or Transformer to model the relationship between each prototype in a set. By exploiting the relationship between prototypes, we may be able to find a better way to distill common knowledge from the training data. For depth estimation, we believe a future direction of fusing the semantic information and the depth information in a smart way may boost the performance since both sources contain geometry information and can be cross-verified during training. As for the future direction on few-shot segmentation, we believe that exploiting the information in unlabeled data will lead to a new state-of-the-art in this field. Currently, the support set only contains a few labeled samples, and the knowledge that the model learned from those is limited. With the fast development of self-supervised methods,e.g, SimCLR [14], how to use that in few-shot learning still remains an interesting and open problem.

Bibliography

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[2] Shuang Ao, Xiang Li, and Charles X Ling. Fast generalized distillation for semi-supervised domain adaptation. In *AAAI 2017*. 13, 20

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 48

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010. 19, 33

[5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *NeurIPS 2016*. 19, 33

[6] Anselm Brachmann, Erhardt Barth, and Christoph Redies. Using cnn features to better understand what makes visual artworks special. *Frontiers in psychology*, 8:830, 2017. vi, 9

[7] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787, 2009. 2

[8] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. vi, 31, 40

[9] Yue Cao, Mingsheng Long, and Jianmin Wang. Unsupervised domain adaptation with distribution matching machines. In *AAAI 2018*. 18, 32

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets,

atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 3, 45

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 48

[12] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 3

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML 2020*. 61

[15] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29:730–738, 2016. 4, 34

[16] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *CVPR*, 2019. 34

[17] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[18] Wen-Sheng Chu, Fernando De la Torre, and Jeffery F Cohn. Selective transfer machine for personalized facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3515–3522, 2013. 2

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 3

[20] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018. 2

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[22] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR 2013*. 1, 13, 20

[23] Jiahua Dong, Yang Cong, Gan Sun, Yuyang Liu, and Xiaowei Xu. Cscl: Critical semantic-consistent learning for unsupervised domain adaptation. In *ECCV 2020*. 18, 32

[24] Jiahua Dong, Yang Cong, Gan Sun, Bineng Zhong, and Xiaowei Xu. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *CVPR*, 2020. 18, 19, 33

[25] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, page 4, 2018. 14, 15, 49

[26] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 4, 33

[27] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 4, 33

[28] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640, 2017. 2

[29] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2423–2436, 2019. 4, 34

[30] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009. 56

[31] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 3

[32] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 47, 48

[33] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Neural Information Processing Systems*, pages 9516–9527, 2018. 48

[34] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 4, 33

[35] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988. 8

[36] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 2

[37] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 41

[38] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML 2015*. 1, 18, 26, 32

[39] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR 2016*. 19

[40] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 4, 34

[41] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. vi, 31

[42] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 40

[43] Marzieh Gheisari and Mahdieh Soleymani Baghshah. Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing*, 165:300–311, 2015. 2

[44] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV 2015*. 19, 33

[45] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016. 19, 33

[46] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020. 34

[47] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 8

[48] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. 4, 14, 34, 39

[49] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. vi, 4, 13, 14, 34, 37, 39, 40, 41, 42, 43

[50] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. In *CAP*, pages 281–296, 2005. 26

[51] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *NeurIPS*, 2006. 19, 33

[52] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR 2012*. 23

[53] Gewen He, Xiaofeng Liu, Fangfang Fan, and Jane You. Classification-aware semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 964–965, 2020. 2

[54] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR 2020*. 20, 22

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR 2016*. 9, 10

[56] Michael Heath, Kevin Bowyer, Daniel Kopans, P Kegelmeyer, Richard Moore, Kyong Chang, and S Munishkumaran. Current status of the digital database for screening mammography. In *Digital mammography*, pages 457–460. Springer, 1998. 10

[57] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 21

[58] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML 2018*. 18, 19, 30, 33

[59] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 43

[60] Tao Hu, Pengwan, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees G. M. Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. 2018. 14, 15, 48

[61] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. *IJCAI 2020*. 13, 18, 19, 20, 33

[62] Xufeng Jiang, Nan Zhou, and Xiang Li. Few-shot segmentation of remote sensing images using deep metric learning. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 46, 49, 55, 56, 57, 58

[63] Dongki Jung, Jaehoon Choi, Yonghan Lee, Deokhwa Kim, Changick Kim, Dinesh Manocha, and Donghwan Lee. Dnd: Dense depth estimation in crowded dynamic indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12797–12807, 2021. 4, 33, 34

[64] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR 2019*. 2, 19, 33

[65] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV 2020*. 17, 18, 19, 20, 22, 23, 24, 25, 26, 28

[66] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 48

[67] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. *Advances in neural information processing systems*, 31, 2018. 48

[68] Simon AA Kohl, Bernardino Romera-Paredes, Klaus H Maier-Hein, Danilo Jimenez Rezende, SM Eslami, Pushmeet Kohli, Andrew Zisserman, and Olaf Ronneberger. A hierarchical probabilistic u-net for modeling multi-scale ambiguities. *arXiv preprint arXiv:1905.13077*, 2019. 48

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS 2012*. 1, 8, 9, 10

[70] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *WACV*, 2021. vii, 4, 31, 33, 34, 40, 41, 42, 43

[71] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8

[72] Seunghun Lee, Sunghyun Cho, and Sunghoon Im. Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15252–15261, June 2021. 33

[73] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1482–1491, June 2021. 33

[74] Gongbo Liang, Xiaoqin Wang, Yu Zhang, Xin Xing, Hunter Blanton, Tawfiq Salem, and Nathan Jacobs. Joint 2d-3d breast cancer classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 692–696. IEEE, 2019. 10

[75] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 48

[76] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[77] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10995, 2019. 4, 34

[78] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NeurIPS*, 2016. 19, 33

[79] Qun Liu and Supratik Mukhopadhyay. Unsupervised learning using pretrained cnn and associative memory bank. In *IJCNN*. IEEE, 2018. 20

[80] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 3, 8, 45, 48, 56

[81] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017. 26

[82] Adrian Lopez-Rodriguez and Krystian Mikolajczyk. Desc: Domain adaptation for depth estimation via semantic consistency. *BMVC*, 2020. 4, 30, 31, 33, 34, 42

[83] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007. 8

[84] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (TOG)*, 39(4):71–1, 2020. 4, 34

[85] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS journal of photogrammetry and remote sensing*, 165:108–119, 2020. vi, 31, 41

[86] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2020. 54, 56

[87] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019. 4, 34

[88] Logambal Madhuanand, Francesco Nex, and Michael Ying Yang. Self-supervised monocular depth estimation from oblique uav videos. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176:1–14, 2021. 4, 34, 41, 54

[89] Claudio Michaelis, Matthias Bethge, and Alexander S Ecker. One-shot segmentation in clutter. *International Conference on Machine Learning*, 2018. 48

[90] Saeid Motiian, Quinn Jones, Seyed Mehdi Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. *arXiv preprint arXiv:1711.02536*, 2017. 19

[91] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR 2021*. 26

[92] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 3

[93] Le Thanh Nguyen-Meidine, Atif Belal, Madhu Kiran, Jose Dolz, Louis-Antoine Blais-Morin, and Eric Granger. Unsupervised multi-target domain adaptation through knowledge distillation. In *WACV*, 2021. 31, 34, 41

[94] Ishan Nigam, Chen Huang, and Deva Ramanan. Ensemble knowledge transfer for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1499–1508. IEEE, 2018. 41, 55

[95] Ishan Nigam, Chen Huang, and Deva Ramanan. Ensemble knowledge transfer for semantic segmentation. 2018. 54, 56

[96] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 8

[97] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020. 30

[98] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2239–2247, 2019. 2

[99] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshop*, 2017. 41

[100] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 25

[101] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *CVPR 2019*. 25

[102] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *ICML*, 2019. 34

[103] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. *arXiv preprint arXiv:1806.07373*, 2018. 15

[104] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018. 14, 15, 48, 52

[105] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 47, 48

[106] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younes Bennani. *Advances in domain adaptation theory*. Elsevier, 2019. 19, 33

[107] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 8

[108] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241, 2015. 48

[109] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1):293–298, 2012. 54

[110] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *TPAMI*, 2018. 19, 33

[111] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *CVPR 2019*. 2, 13, 17, 18, 19, 20, 22, 23, 25, 26, 32, 33

[112] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*, 2017. 26

[113] Antoine Saporta, Tuan-Hung Vu, Mathieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. In *ICCV*, 2021. 31, 32, 35, 41

[114] Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. In *ICCV*, 2021. 32, 34, 40, 42

[115] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 14, 15, 48, 52

[116] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3358–3369, 2019. 2

[117] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 15, 47, 48, 49

[118] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Neural Information Processing Systems*, pages 3483–3491, 2015. 48

[119] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI 2016*. 19, 33

[120] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016*. 19, 33

[121] Takeshi Teshima, Issei Sato, and Masashi Sugiyama. Few-shot domain adaptation by causal mechanism transfer. In *ICML 2020*. 19

[122] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 1

[123] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR 2008*. 28

[124] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR 2017*. 25

[125] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019. 13

[126] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 47, 48

[127] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 2017. 21

[128] Haochen Wang, Yandan Yang, Xianbin Cao, Xiantong Zhen, Cees Snoek, and Ling Shao. Variational prototype inference for few-shot semantic segmentation. In *Winter Conference on Applications of Computer Vision*, pages 525–534, 2021. 52

[129] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019. 15, 46, 49, 51, 55, 56, 58

[130] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018. 19, 33

[131] Xiaoqin Wang, Gongbo Liang, Yu Zhang, Hunter Blanton, Zachary Bessinger, and Nathan Jacobs. Inconsistent performance of deep learning models on mammogram classification. *Journal of the American College of Radiology*, 2020. 10

[132] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020. 19, 33

[133] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR 2018*. 20, 22

[134] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018. 4, 34

[135] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2811–2820, 2019. 4, 34

[136] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR 2015*. 1, 13, 20

[137] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018. 4, 34

[138] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 45, 48

[139] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017. 3

[140] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 13

[141] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *International Conference on Computer Vision*, pages 1685–1694, 2019. 48

[142] Xiaofei Zhang, Yi Zhang, Erik Y. Han, Nathan Jacobs, Qiong Han, Xiaoqin Wang, and Jinze Liu. Classification of whole mammogram and tomosynthesis images using deep convolutional neural networks. *IEEE Transactions on NanoBioscience*, PP:1–1, 06 2018. 10

[143] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint arXiv:1810.09091*, 2018. 14, 15, 16, 48

[144] Yu Zhang, Gongbo Liang, Nathan Jacobs, and Xiaoqin Wang. Unsupervised domain adaptation for mammogram image classification: A promising tool for model generalization. *arXiv preprint arXiv:2003.01111*, 2020. 10

[145] Yu Zhang, Gongbo Liang, Tawfiq Salem, and Nathan Jacobs. Defense-pointnet: Protecting pointnet against adversarial attacks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5654–5660. IEEE, 2019. 2

[146] Yu Zhang, Xiaoqin Wang, Hunter Blanton, Gongbo Liang, Xin Xing, and Nathan Jacobs. 2d convolutional neural networks for 3d digital breast tomosynthesis classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1013–1017. IEEE, 2019. 10

[147] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3, 48

[148] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *CVPR*, 2019. 31, 42

[149] Xiantong Zhen, Yingjun Du, Huan Xiong, Qiang Qiu, Cees Snoek, and Ling Shao. Learning to learn variational semantic memory. *Neural Information Processing Systems*, 2020. 48

[150] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 3

[151] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 4, 34, 40

[152] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 10, 18, 19, 33

[153] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *AAAI 2019*. 23

[154] Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. Learning ordinal relationships for mid-level vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 388–396, 2015. 4, 34

[155] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018. 1

Vita

# Yu Zhang

Education

- B.E. in Telecommunications, Northeastern University (CN).
  Sep. 2013 - June 2017

Professional Appointments

- Image Analytics Intern @ Siemens Healthineers (May 2022 - Aug. 2022)
  Malvern, Pennsylvania.
- Teaching Assistant @ University of Kentucky (Jan. 2018 - May 2021)
  Lexington, Kentucky.
- Research Assistant @ University of Kentucky (May 2019 - Dec. 2022)
  Lexington, Kentucky.

Publications

[1] **Yu Zhang**, M. Rafique, G. Christie, N. Jacobs. “CrossAdapt: Cross-Scene Adaptation for Multi-Domain Depth Estimation”. Under Review.

[2] **Yu Zhang**, M. Rafique, N. Jacobs. “CrossSeg: Cross-Scene Few-Shot Aerial Segmentation Using Probabilistic Prototypes”. Under Review.

[3] **Yu Zhang**, G. Liang, N. Jacobs. “Dynamic Feature Alignment for Semi-supervised Domain Adaptation”. In *British Machine Vision Conference (BMVC)*, 2021. [link]

[4] **Yu Zhang**, G. Liang, Y. Su, N. Jacobs. “Multi-Branch Attention Networks for Classifying Galaxy Clusters”. In *International Conference on Pattern Recognition (ICPR)*, 2020. [link]

[5] **Yu Zhang**, X. Wang, H. Blanton, G. Liang, X. Xing, N. Jacobs. “2D Convolutional Neural Networks for 3D Digital Breast Tomosynthesis Classification”. In *IEEE International Conference of Bioinformatics and Biomedicine (BIBM)*, 2019. [link]

[6] **Yu Zhang**, G. Liang, N. Jacobs, X. Wang. “Unsupervised Domain Adaptation for Mammogram Image Classification: A Promising Tool for Model Generalization”. In *Conference on Machine Intelligence in Medical Imaging*, 2019. [link]

[7] **Yu Zhang**, G. Liang, T. Salem, N. Jacobs. “Defense-PointNet: Protecting PointNet Against Adversarial Attacks”. In *IEEE BigData Workshop: The Next Frontier of Big Data From LiDAR*, 2019. [link]

[8] X. Xing, M. Rafique, G. Liang, G. Blanton, **Yu Zhang**, C. Wang, N Jacobs, A. Lin. “Efficient Training on Alzheimer’s Disease Diagnosis with Learnable Weighted Pooling for 3D PET Brain Image Classification”. In *Electronics*, 2023. [link]

[9] X. Xing, C. Peng, **Yu Zhang**, A. Lin, N. Jacobs. “AssocFormer: Association Transformer on Multi-label Classification”. In *British Machine Vision Conference (BMVC)*, 2022. [link]

[10] X. Xing, G. Liang, **Yu Zhang**, S. Khanal, A. Lin, N. Jacobs. “ADVIT: Vision Transformer on Multi-modality PET Images for Alzheimer Disease Diagnosis”. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2022. [link]

[11] S. Lin, Y. Su, G. Liang, Y. Zhang, N. Jacobs, **Yu Zhang**. “Estimating Cluster Masses from SDSS Multi-band Images with Transfer Learning”. In *Monthly Notices of the Royal Astronomical Society (MNRAS)*, 2022. [link]

[12] G. Liang, C. Greenwell, **Yu Zhang**, X. Wang, R. Kavuluru, N. Jacobs. “Contrastive Cross-Modal Pre-Training: A General Strategy for Small Sample Medical Imaging”. In *IEEE Journal of Biomedical and Health Informatics (JBHI)*, 2021. [link]

[13] Y. Su, **Yu Zhang**, G. Liang, J. A. ZuHone, D. J. Barnes, N. B. Jacobs, M. Ntampaka, W. R. Forman, R. P. Kraft, P. E. J. Nulsen, C. Jones, E. Roediger. “A deep learning view of the census of galaxy clusters in IllustrisTNG”. In *Monthly Notices of the Royal Astronomical Society (MNRAS)*, 2020. [link]

[14] X. Wang, G. Liang, **Yu Zhang**, H. Blanton, Z. Bessinger, N. Jacobs. “Inconsistent Performance of Deep Learning Models on Mammogram Classification”. In *Journal of the American College of Radiology (JACR)*, 2020. [link]

[15] G. Liang, X. Xing, L. Liu, **Yu Zhang**, Q. Ying, A. Lin, and N. Jacobs. "Alzheimer's Disease Classification Using 2D Convolutional Neural Networks". In *IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021. [link]

[16] G. Liang, **Yu Zhang**, X. Wang, N. Jacobs. "Improved Trainable Calibration Method for Neural Networks on Medical Imaging Classification". In to *British Machine Vision Conference (BMVC)*, 2020. [link]

[17] G. Liang, X. Wang, **Yu Zhang**, N. Jacobs. "Weakly-Supervised Self-Training for Breast Cancer Localization". In *IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020. [link]

[18] U. Rafique, **Yu Zhang**, B. Brodie, N. Jacobs. "Unifying Guided and Unguided Outdoor Image Synthesis". In *CVPR Workshop*: NTIRE 2021. [link]

[19] G. Liang, S. Lin, **Yu Zhang**, Y. Su, Nathan Jacobs. "Optical Wavelength Guided Self-Supervised Feature Learning For Galaxy Cluster Richness Estimate". In *NeurIPS Workshop: Machine Learning and Physical Sciences*, 2020. [link]

[20] G. Liang, **Yu Zhang**, N. Jacobs. "Neural Network Calibration for Medical Imaging Classification Using DCA Regularization". In *ICML Workshop: Uncertainty and Robustness in Deep Learning*, 2020. [link]

[21] G. Liang, X. Wang, **Yu Zhang**, X. Xing, H. Blanton, T. Salem, N. Jacobs. "Joint 2D-3D Breast Cancer Classification". In *IEEE International Conference of Bioinformatics and Biomedicine (BIBM)*, 2019. [link]

[22] G. Liang, **Yu Zhang**, J. Liu, N. Jacobs, X. Wang. "Training Deep Learning Models as Radiologists: Breast Cancer Classification Using Combined Whole 2D Mammography and Full Volume Digital Breast Tomosynthesis". In *Radiological Society of North America 105th Scientific Assembly and Annual Meeting*, 2019. [link]

Professional Service

- Reviewer for IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
- Reviewer for IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)

- Reviewer for IEEE Winter Conference on Applications of Computer Vision 2020, 2022
- Reviewer for The British Machine Vision Conference 2020, 2021
- Reviewer for Imaging Science Journal 2022