UNSUPERVISED DOMAIN ADAPTATION WITH DEEP NEURAL NETWORKS

by

JINYU YANG

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2022

To my parents and my brother for their endless trust, support, and love.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervising professor, Dr. Junzhou Huang who inspired me to do this dissertation without whom this dissertation would not have been possible. His irreplaceable encouragement and supervision are the main reasons for the successful outcomes of my research. Dr. Huang's incredible breadth and depth of knowledge, creative research ideas, constructive, insightful, and encouraging conversations inspire me in both research and life. Over the past three years, I have learned a lot from Dr. Huang, including but not limited to: how to effectively and efficiently read a paper, how to come up with new ideas, how to write an informative and well-written paper, and how to build connections with other researchers. I believe all of them will also play an important role and guidance in my future career.

I sincerely express my gratitude to Dr. Dajiang Zhu, Dr. Jia Rao, and Dr. Yingying Zhu for serving on my committee. I have benefited a lot from their invaluable suggestions and feedback on my diagnostic evaluation, comprehensive exam, proposal defense, and dissertation. Furthermore, they have long inspired me through their contributions to operating systems, medical image analysis, and deep learning.

For three years, it is my pleasure to work with lab members from the Scalable Modeling & Imaging & Learning Lab (SMILE), including Zheng Xu, Ruoyu Li, Jiawen Yao, Xinliang Zhu, Sheng Wang, Mohammad Minhazul Haq, Chaochao Yan, Ashwin Raju, Yuzhi Guo, Hehuan Ma, Zeheng Li, Chunyuan Li, Weizhi An, Xinsheng Li, Feng Tong, Saiyang Na, and Wenliang Zhong. I want to thank all of them for their insightful discussion and collaboration.

I deeply appreciate the guidance from Peilin Zhao, Yu Rong, Ying Wei, TingYang Xu and Wenbing Huang from Tencent AI Lab, Jingjing Liu and Ning Xu from Kuaishou Technology, and Son Tran, Jiali Duan, Yi Xu, Sampath Chanda and Liqun Chen from Amazon.

I was extremely fortunate to have received invaluable instructions from Dr. Bahram Khalili, as well as assistance and help from other CSE staff members, especially Ginger Dickens, Sha'Londa Towns, and Pamela Mcbride.

Finally, I thank my parents and my brother, for their unconditional love.

April 26, 2022

ABSTRACT

UNSUPERVISED DOMAIN ADAPTATION WITH DEEP NEURAL NETWORKS

Jinyu Yang, Ph.D.

The University of Texas at Arlington, 2022

Supervising Professor: Dr. Junzhou Huang

Deep neural networks (DNNs) demonstrate unprecedented achievements on various machine learning problems and applications. However, such impressive performance heavily relies on massive amounts of labeled data which requires considerable time and labor efforts to collect and annotate. To remedy this limitation, unsupervised domain adaptation (UDA) has attracted more and more attention in the past decade, owing to its capability in transferring the knowledge learned from a labeled source domain to an unlabeled target domain. UDA has proved its wide applicability in various vision tasks, for example, image classification and semantic segmentation. Despite its impressive success, the limitations of existing UDA methods lie in that: i) the consistency of the joint distribution in the target domain cannot be guaranteed by simply performing global feature alignment as in previous studies; ii) the context-dependency is essential for semantic segmentation, however, its transferability is still not well understood; iii) the robustness of UDA methods in semantic segmentation remains unexplored, which poses a security concern in this field; and iv) previous work is mainly built upon convolutional neural networks (CNNs) to learn domain-invariant

representations. However, the transferability of the Vision Transformer (ViT) which is convolution-free, is still an open problem.

To address these limitations, in this dissertation: i) we use a reconstruction network to reconstruct both source and target images from their predicted labels. Therefore, we can encourage cross-domain features with the same category close to each other; ii) we design two cross-domain attention modules to adapt context dependencies from both spatial and channel views. Specifically, the spatial attention module captures local feature dependencies between each position in the source and target image. The channel attention module models semantic dependencies between each pair of cross-domain channel maps. In consequence, the contextual information can be aggregated and adapted across domains; iii) we comprehensively evaluate the robustness of existing UDA methods and propose a robust UDA approach that maximizes the agreement between clean images and their adversarial examples by a contrastive loss in the output space. iv) we perform the first-of-its-kind investigation of ViT's generalization ability on commonly used benchmarks and propose a new UDA method that explicitly considers the intrinsic merits of the transformer architecture.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

Introduction

With the recent exponential increase in large-scale datasets, deep neural networks (DNNs) [1] demonstrate their impressive power in various tasks and applications, such as image recognition [2], language understanding [3], and regulatory genomics [4, 5, 6]. For example, massive amounts of videos, images, and texts can be easily accessed on the Internet, which provides a unique opportunity to train deeper and more powerful neural networks [7]. Notwithstanding, it is widely recognized that DNNs heavily rely on massive labeled data which might be infeasible in practice [8]. One typical example is the medical data, which is hard to collect and requires massive amounts of labor efforts in label annotation [9, 10, 11]. Therefore, it is desirable to train models that can leverage rich labeled data from a different but related domain and generalize well on the domain of interest. Unfortunately, the canonical supervised-learning paradigm suffers from the domain shift issue that poses a major challenge in adapting models across domains. To address this limitation, transfer learning [8] attracts considerable attention in the past few decades. The key idea of transfer learning is to leverage knowledge from a labeled source domain to effectively learn a model in a target domain that has limited labeled data. The wide applicability of transfer learning has been proved in, for example, image classification [12, 13, 13], objection detection [14, 15, 16], semantic segmentation [17, 18, 19, 20, 21, 22], and NLP [23]. In this dissertation, we focus on unsupervised domain adaptation (UDA) which is a special case of transfer learning. Specifically, UDA refers to the scenario where the target domain is totally unlabeled.

Algorithm		S→M	U→M	M→U	Avg
Source Only	LeNet	67.1	69.6	82.2	73.0
RevGrad [24]		73.9	73.0	77.1	74.7
ADDA [25]		76.0	90.1	89.4	85.2
Target Only		99.4	99.4	98.0	98.9

Table 1.1: Merits of transfer learning on MNIST (M), USPS(U), and SVHN(S).

In this chapter, we first introduce the motivation of this study in Section 1.1. After that, we formally introduce UDA and related works in Section 1.2 and Section 1.3, respectively. We then point out the research challenges in this field in Section 1.4. For these challenges, we highlight our contributions in Section 1.5. In the end, we introduce the structure of this dissertation in Section 1.6.

1.1 Motivation

Although DNNs demonstrate record-breaking performance under the supervised learning paradigm, they suffer from domain shift issues of cross-domain mismatches in feature space, distribution, label space, and predictive model. Such domain discrepancy results in dramatic performance degradation when directly applying models learned from one domain to another domain. To show the motivation of this study, we report some preliminary results on the digit recognition task with three datasets, i.e., MNIST [26], USPS, and Street View House Numbers (SVHN) [27], where each dataset can be regarded as a domain. Therefore, a total of three source-target domain pairs are available, e.g., S→M indicates that SVHN is the source domain and MNIST is the target domain. For the backbone LeNet [26], we report its lower bound performance (73.0% on average) denoted by Source Only, meaning the model is trained with source data only. We also show the Target Only results as the upper bound performance (98.9% on average), which is obtained by both training and testing on the labeled target data. As shown in Table1.1, there is a large performance gap (25.9% on average)

between Source Only and Target Only, revealing the defect of failing to consider the domain discrepancy. By contrast, two transfer learning methods [24, 25] significantly reduce the performance gap by leveraging the cross-domain knowledge, indicating the necessity of this research topic. However, existing transfer learning methods still lag behind Target Only by a large margin, we, therefore, hope to design new algorithms from innovative perspectives to further facilitate this area.

1.2 Unsupervised Domain Adaptation

Before diving into UDA, we first introduce two fundamental concepts in transfer learning, i.e., "domain" and "task". Formally, i) a domain is denoted by $\mathcal{D} = \{\mathcal{X}, P(X)\}$ which contains two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$. Specifically, $X = \{x_1, ..., x_n\} \in \mathcal{X}$. ii) given a specific domain $\mathcal{D}$, a task is defined as $\mathcal{T} = \{\mathcal{Y}, P(y|x)\}$ which contains a label space $\mathcal{Y}$ and a conditional probability distribution $P(y|x)$. Based on these two concepts, transfer learning is defined as follows.

Definition 1.2.1 (Transfer Learning) *Given a source domain $\mathcal{D}_S = \{\mathcal{X}_S, P_S(X)\}$ associated with its learning task $\mathcal{T}_S = \{\mathcal{Y}_S, P_S(y|x)\}$, and a target domain $\mathcal{D}_T = \{\mathcal{X}_T, P_T(X)\}$ together with its learning task $\mathcal{T}_T = \{\mathcal{Y}_T, P_T(y|x)\}$, transfer learning aims to reduce the generalization error on the target domain where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

Based on the difference between the feature space, marginal distribution, label space, and conditional distribution, transfer learning can be further categorized as homogeneous transfer learning and heterogeneous transfer learning. Specifically, homogeneous transfer learning assumes that $\mathcal{X}_S = \mathcal{X}_T$ and $\mathcal{Y}_S = \mathcal{Y}_T$, while $P_S(X) \neq P_T(X)$ or $P_S(y|x) \neq P_T(y|x)$. Heterogeneous transfer learning relaxes the assumption

in homogeneous transfer learning to allow that $\mathcal{X}_S \neq \mathcal{X}_T$ or $\mathcal{Y}_S \neq \mathcal{Y}_T$. UDA falls into the homogeneous transfer learning by further assuming that $P_S(y|x) = P_T(y|x)$.

The most commonly used UDA algorithms are based on feature representation learning which attempts to learn new representations $\phi(x)$ to minimize the difference between $P_S(\phi(x))$ and $P_T(\phi(x))$. The assumption of feature representation learning is: some features in the feature space are shared by the source and the target domain, while the others are domain-specific, or there exists a hidden feature space that is shared by two domains. One common practice in feature representation learning is probability distribution adaptation, which explicitly maximizes the distribution similarity. This is achieved by learning a new representation $\phi(x)$ through explicitly minimizing the distance between $P_S(\phi(x))$ and $P_T(\phi(x))$.

1.3 Existing Methods

In the past few decades, various UDA methods have been proposed for image classification tasks [12, 13, 28]. For instance, DDC [12] attempts to learn domain-invariant features by minimizing Maximum Mean Discrepancy (MMD) [29] between two domains. Long et al. further improve DDC by embedding hidden representations of all task-specific layers in a reproducing Hilbert space and use a multiple kernel variant of MMD to measure the domain distance [13]. JAN [28] aligns joint distributions of multiple domain-specific layers across domains through a joint maximum mean discrepancy metric. Another line of effort was inspired by the success of adversarial learning [30]. By introducing a domain discriminator and modeling the domain adaption as a minimax problem [24, 25, 31], an encoder is trained to generate domain-invariant features, through deceiving a discriminator which tries to distinguish features of the source domain from that of the target domain. Therefore, the features learned

by the encoder are both transferable across domains and discriminative for downstream tasks.

Semantic segmentation, or image segmentation, aims to predict pixel-level labels for the given images. Since it plays a fundamental role in, for example, autonomous driving, there is surging interest in designing UDA methods for semantic segmentation tasks. These methods can be summarized as four streams: i) adapt domain-invariant features by directly minimizing the representation distance between two domains [17, 32]; ii) align pixel space through translating images from the source domain to the target domain [18, 33]; iii) align structured output space, which is inspired by the fact that source output and target output share substantial similarities in terms of structure layout [19]; iv) generate pseudo labels for target images and then re-training the segmentation model with these labels [34, 35, 36, 37, 38, 39].

1.4 Research Challenges

Although existing UDA methods have proved their effectiveness in various domain adaptation tasks, they still suffer from the following challenges. **Challenge 1**: simply maximizing the marginal distribution similarity through Maximum Mean Discrepancy (MMD) ignores the joint distribution shift. This limitation gives rise to severe false positive and false negative issues in the target prediction. This problem can get even worse when there is a significant discrepancy in layout or structure between the source and target domains, such as adapting from synthetic to real urban traffic scenes. **Challenge 2**: existing methods fail to explicitly consider the contextual dependencies across the source and target domains which is essential for scene understanding. **Challenge 3**: like other machine learning methods, UDA methods are also possibly vulnerable to adversarial attacks. However, the robustness of UDA methods remains largely unexplored in the literature. With the increasing

applications of UDA methods in security-related areas, the lack of robustness of these methods leads to massive safety concerns. **Challenge 4**: despite that ViT [40] is becoming increasingly popular, two important questions related to domain adaption remain unanswered. First, *how does the generalization ability of ViT across different domains?* There are several contemporary work [41, 42, 43] that apply DeiT [44] and Swin [45] to UDA, yet the ViT has not been investigated. The second question is, *how can we properly improve ViT in adapting different domains?* One intuitive approach is to directly apply adversarial discriminator onto the class tokens to perform adversarial alignment, where the state of a class token represents the entire image. However, cross-domain alignment of such global features assumes all regions or aspects of the image have the equal transferability and discriminative potential, which is not always tenable.

1.5 Contributions

In this dissertation, we pursue to address the aforementioned research challenges via innovative research. For **Challenge 1**, we propose a label-driven reconstruction network [46] which reconstructs both source and target images from their predicted semantic labels. This is essential to guide the segmentation network by penalizing the reconstructed image that semantically deviates from the corresponding input image. Most importantly, this strategy enforces cross-domain features with the same category close to each other. For **Challenge 2**, we design a cross-attention mechanism that contains two cross-domain attention modules to capture mutual context dependencies between source and target domains [20]. Given that same objects with different appearances and scales often share similar features, we introduce a cross-domain spatial attention module (CD-SAM) to capture local feature dependencies between any two positions in a source image and a target image. To model the associations

between different semantic responses across two domains, we introduce a cross-domain channel attention module (CD-CAM) which has the same bidirectional structure as CD-SAM. For **Challenge 3**, we first perform a comprehensive study to evaluate the robustness of existing UDA methods in semantic segmentation[21]. We then introduce a new UDA method to robustly adapt domain knowledge in urban-scene semantic segmentation. The key insight is to leverage the regularization power of adversarial examples. For **Challenge 4**, we first comprehensively investigates the performance of ViT on a variety of domain adaptation tasks. To further improve the power of ViT in transferring domain knowledge, we propose TVT by explicitly considering the intrinsic merits of transformer architecture. Specifically, TVT captures both transferable and discriminative features in the given image, and retains discriminative information of the learnt domain-invariant representations.

1.6 Dissertation Structure

This dissertation is organized as follows. In Chapter 2, we learn how to use semantic label information to facilitate UDA in segmentation tasks. This chapter is primarily based on [46]. In Chapter 3, we propose an innovative cross-attention mechanism for domain adaptation by adapting the semantic context. This chapter is primarily based on [20]. In Chapter 4, we empirically reveal that existing UDA methods can be easily deceived by unnoticeable perturbations and propose a new model to improve the model's robustness. This chapter is primarily based on [21]. In Chapter 5, we first comprehensively investigates the generalization ability of ViT on a variety of domain adaptation tasks. Then we propose an unified framework, namely Transferable Vision Transformer (TVT), to fully exploit the transferability of ViT for domain adaptation. Specifically, we delicately devise a novel and effective unit, which we term Transferability Adaption Module (TAM). By injecting learned

transferabilities into attention blocks, TAM compels ViT focus on both transferable and discriminative features. Besides, we leverage discriminative clustering to enhance feature diversity and separation which are undermined during adversarial domain alignment. This chapter is primarily based on [22]. We conclude and discuss future directions in Chapter 6.

CHAPTER 2

Label-Driven Reconstruction for Domain Adaptation in Semantic Segmentation

In this chapter, we study how to address two common limitations of existing UDA methods in semantic segmentation. The first limitation is introduced by the image-to-image translation strategy which translates images from the source domain to the target domain. Although this strategy reduces the appearance discrepancy between two domains, source-to-target translation enlarges the bias in translated images and introduces extra computations, owing to the dominant data size of the source domain. The second limitation is that the consistency of joint distributions in source and target domains cannot be guaranteed through global feature alignment. Here, we present an innovative framework, designed to mitigate the image translation bias and align cross-domain features with the same category. This is achieved by 1) performing the target-to-source translation and 2) reconstructing both source and target images from their predicted labels. Extensive experiments on adapting from synthetic to real urban scene understanding demonstrate that our framework competes favorably against existing state-of-the-art methods.

2.1 Introduction

Deep Convolutional Neural Networks (DCNNs) have demonstrated impressive achievements in computer vision tasks, such as image recognition [7], object detection [47], and semantic segmentation [48]. As one of the most fundamental tasks, semantic

segmentation predicts pixel-level semantic labels for given images. It plays an extremely important role in autonomous agent applications such as self-driving techniques.

Existing supervised semantic segmentation methods, however, largely rely on pixel-wise annotations which require tremendous time and labor efforts. To overcome this limitation, publicly available synthetic datasets (e.g., GTA [49] and SYNTHIA [50]) which are densely-annotated, have been considered recently. Nevertheless, the most obvious drawback of such a strategy is the poor knowledge generalization caused by domain shift issues (e.g., appearance and spatial layout differences), giving rise to dramatic performance degradation when directly applying models learned from synthetic data to real-world data of interest. In consequence, domain adaptation has been exploited in recent studies for cross-domain semantic segmentation, where the most common strategy is to learn domain-invariant representations by minimizing distribution discrepancy between source and target domains [51, 52], designing a new loss function [32], considering depth information [53, 54], or alternatively generating highly confident pseudo labels and re-training models with these labels through a self-training manner [55, 34, 35, 36, 37, 38, 39]. Following the advances of Generative Adversarial Nets (GAN) [30], adversarial learning has been used to match cross-domain representations by minimizing an adversarial loss on the source and target representations [17, 56, 57, 58], or adapting structured output space across two domains [19, 34]. Recent studies further consider the pixel-level (e.g., texture and lighting) domain shift to enforce source and target images to be domain-invariant in terms of visual appearance [59, 18, 60, 61, 62, 20]. This is achieved by translating images from the source domain to the target domain by using image-to-image translation models such as CycleGAN [63] and UNIT [64].

Despite these painstaking efforts, we are still far from being able to fully adapt cross-domain knowledge mainly stemming from two limitations. First, adversarial-

based image-to-image translation introduces inevitable bias to the translated images, as we cannot fully guarantee that the translated source domain $\mathcal{F}(\mathcal{X}_s)$ is identical to the target domain $\mathcal{X}_t$ ($\mathcal{X}_s$ and $\mathcal{X}_t$ denote two domains, and $\mathcal{F}$ indicates an image-to-image translation model). This limitation is especially harmful to the source-to-target translation [59, 18, 60, 61, 34], since the data size of the source domain is much larger than the target domain in most of domain adaptation problems. Moreover, source-to-target translation is more computationally expensive than target-to-source translation. Second, simply aligning cross-domain representations in the feature space [17, 18, 19] ignores the joint distribution shift (i.e., $\mathcal{P}(G(\mathcal{X}_s), Y_s) \neq \mathcal{P}(G(\mathcal{X}_t), Y_t)$, where G is used for feature extraction, while Y_s and Y_t indicate ground truth labels). These limitations give rise to severe false positive and false negative issues in the target prediction. This problem can get even worse when there is a significant discrepancy in layout or structure between the source and target domains, such as adapting from synthetic to real urban traffic scenes.

In this chapter, we propose an innovative domain adaptation framework for semantic segmentation. The key idea is to reduce the image translation bias and align cross-domain feature representations through image reconstruction. As opposed to performing source-to-target translation [18, 60, 34], for the first time, we conduct the target-to-source translation to make target images indistinguishable from source images. This enables us to substantially reduce the bias in translated images and allows us to use original source images and their corresponding ground truth to train a segmentation network. Compared to the source-to-target translation, our method is also much more efficient. Besides, a reconstruction network is designed to reconstruct both source and target images from their predicted labels. It is noteworthy that we reconstruct images directly from the label space, rather than the feature space as reported in previous studies. This is essential to guide the segmentation network by

Figure 2.1: An example of our method on synthetic-to-real urban scene adaptation. Given a target-domain (or real) image (a), we first make target-to-source translation to obtain source-like (or synthetic) image (b), and then perform segmentation on these translated images. Our method improves the segmentation accuracy in the target domain by reconstructing both source and target images from their predicted labels (c). (d) illustrates the image reconstructed from (c), while (e) indicates the ground truth label.

penalizing the reconstructed image that semantically deviates from the corresponding input image. Most importantly, this strategy enforces cross-domain features with the same category close to each other.

The performance of our method is evaluated on synthetic-to-real scenarios of urban scene understanding, i.e., GTA5 to Cityscapes and SYNTHIA to Cityscapes. Our results demonstrate that the proposed method achieves significant improvements compared with existing methods. Figure 2.1 demonstrates an example of our model in adapting cross-domain knowledge in semantic segmentation tasks and reconstructing the input image from its output label. We also carry out comprehensive ablation studies to analyze the effectiveness of each component in our framework.

The contribution of this chapter is threefold.

- For the first time, we propose and investigate the target-to-source translation in domain adaptation. It reduces the image translation bias and is more computationally efficient compared to the widely-used source-to-target translation.
- To enforce semantic consistency, we introduce a label-driven reconstruction module that reconstructs both source and target images from their predicted labels.

- Extensive experiments show that our method achieves the new state-of-the-art performance on adapting synthetic-to-real semantic segmentation.

2.2 Related Work

2.2.1 Semantic Segmentation

Recent achievements in semantic segmentation mainly benefit from the technical advances of DCNNs, especially the emergence of Fully Convolutional Network (FCN) [48]. By adapting and extending contemporary deep classification architectures fully convolutionally, FCN enables pixel-wise semantic prediction for any arbitrary-sized inputs and has been widely recognized as one of the benchmark methods in this field. Numerous methods inspired by FCN were then proposed to further enhance segmentation accuracy, which have exhibited distinct performance improvement on the well-known datasets (e.g., PASCAL VOC 2012 [65] and Cityscapes [66]) [67, 68, 69, 70, 71].

However, such methods heavily rely on human-annotated, pixel-level segmentation masks, which require extremely expensive labeling efforts [66]. In consequence, weakly-supervised methods, which are based on easily obtained annotations (e.g., bounding boxes and image-level tags), were proposed to alleviate the need for effort-consuming labeling [72, 73]. Another alternative is to resort to freely-available synthetic datasets (e.g., GTA5 [49] and SYNTHIA [50]) with pixel-level semantic annotations. However, models learned on synthetic datasets suffer from significant performance degradation when directly applied to the real datasets of interest, mainly owing to the domain shift issue.

2.2.2 Domain Adaptation

Domain adaptation aims to mitigate the domain discrepancy between a source and a target domain, which can be further divided into supervised adaptation, semi-supervised adaptation, and unsupervised adaptation, depending on the availability of labels in the target domain. The term unsupervised domain adaptation refers to the scenario where target labels are unavailable and have been extensively studied [13, 12, 74, 75, 25, 76, 77].

Recent publications have highlighted the complementary role of pixel-level and representation-level adaptation in semantic segmentation [18, 33, 59, 60, 53], where the pixel-level adaptation is mainly achieved by translating images from the source domain to the target domain (source-to-target translation). Specifically, unpaired image-to-image translation is used in CyCADA [18] to achieve pixel-level adaptation by restricting cycle-consistency. Similarly, FCAN achieves the image translation by combining the image content in the source domain and the "style" from the target domain [59]. I2IAdapt [33] further considers to align source and target representations based on an image-to-image translation strategy, attempting to adapt domain shift. Instead of using the adversarial learning for image translation, DCAN performs source-to-target translation by leveraging target images for channel-wise alignment [60]. Driven by the fact that geometry and semantics are coordinated with each other, GIO-Ada augments the standard image translation network by integrating geometric information [53]. However, source-to-target translation introduces substantial bias to the translated images, given that the size of the source domain is usually much larger than the target domain. To address this problem, we propose the first-of-its-kind target-to-source image translation to reduce pixel-level domain discrepancy. Compared to the source-to-target translation, it is more computationally efficient and enables us

to remove the uncertainty by training the segmentation network with original source images and their corresponding labels.

Motivated by the observation that cross-domain images (e.g., GTA5 and Cityscapes) often share tremendous structural similarities, ASN [19] adapts structured output based on the adversarial learning. The strength of this method is its ability to provide weak supervision to target images by enforcing target outs to be indistinguishable from source outputs. However, it is limited to the scenario where two domains have a huge layout discrepancy, resulting in meaningless predictions for target images. To address this limitation, we further enforce the semantic consistency between target images and their predicted labels through a reconstruction network.

Inspired by the self-training, [34, 35, 36, 37, 38, 39] generate pseudo labels for target images and then re-training the segmentation model with these labels. It outperforms the existing methods by a large margin. However, such a strategy underestimates the side effect of pseudo labels that are incorrectly predicted. As a consequence, the segmentation model fails to increasingly improve itself using these wrong ground truth. Instead, our method reconstructs source and target input images from the label space to ensure these outputs are semantically correct. The image-to-image translation network in [34] uses a reconstruction loss and a perceptual loss to maintain the semantic consistency between the input image and the reconstruction from the translated image. Different from [34], we design a cycle-reconstruction loss in our reconstruction network to enforce the semantic consistency between the input image and the reconstruction from the predicted label.

Reconstruction-based strategy for unsupervised domain adaptation has received considerable attention recently [78, 79]. The key idea is to reconstruct input images from their feature representations to ensure that the segmentation model can learn useful information. Chang et al. [80] follow a similar idea to first disentangle images

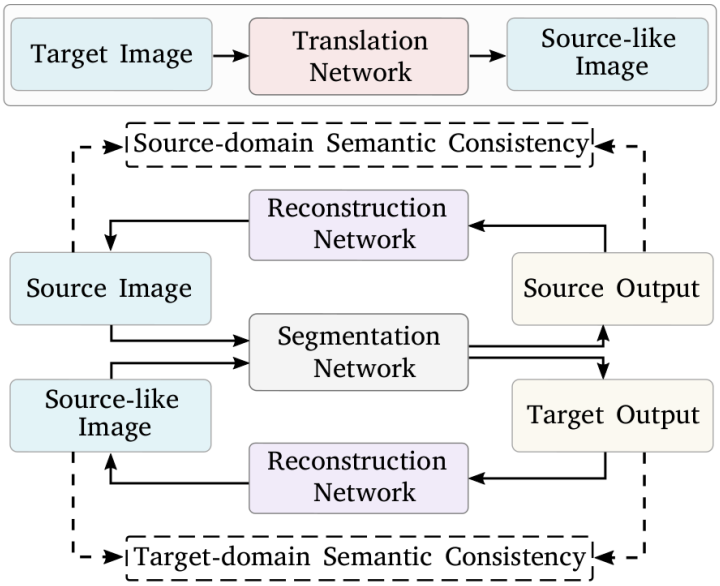

Figure 2.2: An overview of our framework.

into the domain-invariant structure and domain-specific texture representations, and then reconstruct input images. LSD-seg [81] first reconstructs images from the feature space, and then apply a discriminator to the reconstructed images. Rather than performing reconstruction from feature representations, we reconstruct both source and target images from their predicted labels.

2.3 Algorithm

2.3.1 Overview

The overall design of our framework is illustrated in Figure 2.2, mainly containing three complementary modules: a translation network $\mathcal{F}$, a segmentation network G, and a reconstruction network $\mathcal{M}$. Given a set of source domain images $\mathcal{X}_s$ with labels Y_s and a set of target domain images $\mathcal{X}_t$ without any annotations. Our goal is to train G to predict accurate pixel-level labels for $\mathcal{X}_t$. To achieve this, we first use $\mathcal{F}$ to adapt pixel-level knowledge between $\mathcal{X}_t$ and $\mathcal{X}_s$ by translating $\mathcal{X}_t$ to source-like images

$\mathcal{X}_{t\to s}$. This is different from existing prevalent methods that translate images from the source domain to the target domain. $\mathcal{X}_s$ and $\mathcal{X}_{t\to s}$ are then fed into G to predict their segmentation outputs $G(\mathcal{X}_s)$ and $G(\mathcal{X}_{t\to s})$, respectively. To further enforce semantic consistency of both source and target domains, $\mathcal{M}$ is then applied to reconstruct $\mathcal{X}_s$ and $\mathcal{X}_{t\to s}$ from their predicted labels. Specifically, a cycle-reconstruction loss is proposed to measure the reconstruction error, which enforces the semantic consistency and further guides segmentation network to predict more accurate segmentation outputs.

2.3.2 Target-to-source Translation

We first perform the image-to-image translation to reduce the pixel-level discrepancy between source and target domains. As opposed to the source-to-target translation reported in previous domain adaptation methods, we conduct the target-to-source translation through an unsupervised image translation network (Figure 2.3). Our goal is to learn a mapping $\mathcal{F} : \mathcal{X}_t \to \mathcal{X}_s$ such that the distribution of images from $\mathcal{F}(\mathcal{X}_t)$ is indistinguishable from the distribution of $\mathcal{X}_s$. As a counterpart, the inverse mapping $\mathcal{F}^{-1} : \mathcal{X}_s \to \mathcal{X}_t$, which maps images from $\mathcal{X}_s$ to $\mathcal{X}_t$, is introduced to prevent the mode collapse issue [82]. Two adversarial discriminators $\mathcal{D}_t$ and $\mathcal{D}_s$ are employed for distribution match, where $\mathcal{D}_t$ enforces indistinguishable distribution between $\mathcal{F}(\mathcal{X}_t)$ and $\mathcal{X}_s$, and $\mathcal{D}_s$ encourages indistinguishable distribution between $\mathcal{F}^{-1}(\mathcal{X}_s)$ and $\mathcal{X}_t$.

Based on the trained model $\mathcal{F}$, we first translate images from $\mathcal{X}_t$ to source-like images $\mathcal{X}_{t\to s} = \mathcal{F}(\mathcal{X}_t)$. Specifically, each image in $\mathcal{X}_{t\to s}$ preserves the same content as the corresponding image in $\mathcal{X}_t$ while demonstrating the common style (e.g., texture and lighting) as $\mathcal{X}_s$. $\mathcal{X}_s$ and $\mathcal{X}_{t\to s}$ are then fed into a segmentation network for semantic label prediction.

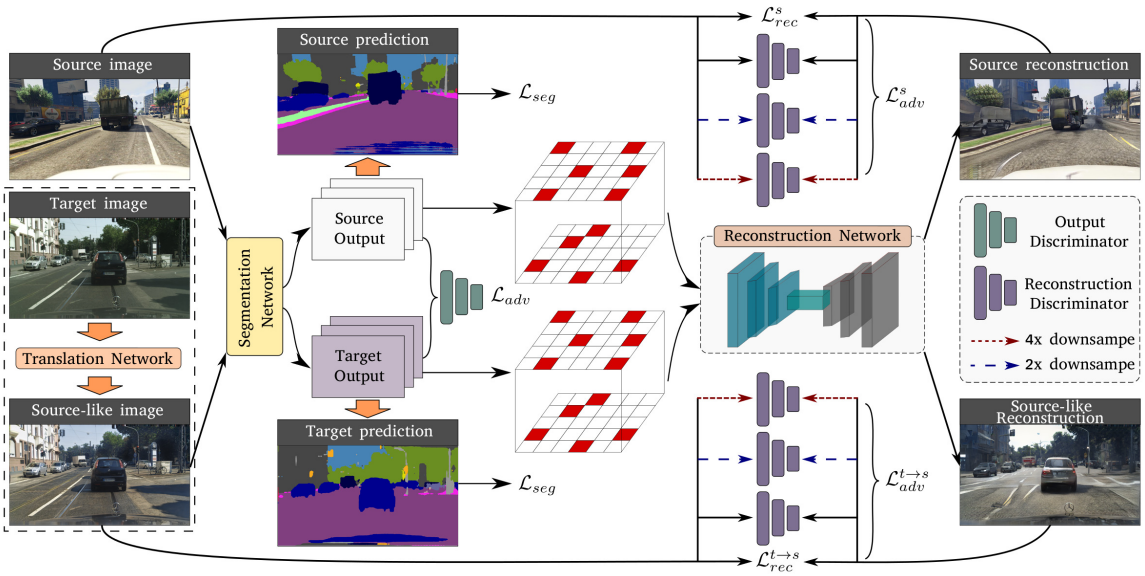

Figure 2.3: Schematic overview of our framework which has three modules: (i) a translation network for pixel-level discrepancy reduction by translating target images to source-like images, where source-like images are indistinguishable from source images, (ii) a segmentation network that predicts segmentation outputs for source images and source-like images, and (iii) a reconstruction network for reconstructing source and source-like images from their corresponding label space.

Compared to translating images from the source domain to the target domain, the target-to-source translation has three benefits. First, it allows full supervision on the source domain by training the segmentation network with original source images and their corresponding labels. Second, it enables to reduce the bias in translated images. Third, it is computationally efficient lying in the fact that $|\mathcal{X}_t| \ll |\mathcal{X}_s|$.

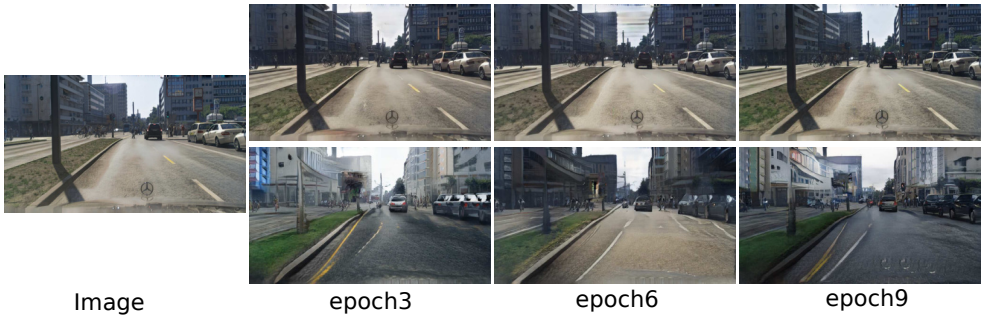

Figure 2.4: A comparison between the image reconstruction from feature space and label space (ours). For each input image (first column), the first and second row indicate the images reconstructed from the feature and label space, respectively.

2.3.3 Semantic Segmentation

Given that source-like images $\mathcal{X}_{t\to s}$ preserves all semantic information from $\mathcal{X}_t$, we apply a shared segmentation network G to $\mathcal{X}_s$ and $\mathcal{X}_{t\to s}$ to predict their segmentation outputs with the loss function given by,

$$\begin{aligned}\mathcal{L}_G = \mathcal{L}_{seg}(G(\mathcal{X}_s), Y_s) + \mathcal{L}_{seg}(G(\mathcal{X}_{t\to s}), Y_t^{ssl}) + \\ \lambda\mathcal{L}_{adv}(G(\mathcal{X}_s), G(\mathcal{X}_{t\to s})),\end{aligned} \tag{2.1}$$

where $\mathcal{L}_{seg}$ indicates the typical segmentation objective, Y_t^{ssl} is pseudo labels of $\mathcal{X}_{t\to s}$ which is derived from [34], $\mathcal{L}_{adv}(G(\mathcal{X}_s), G(\mathcal{X}_{t\to s}))$ is an adversarial loss, and λ leverages the importance of these losses. Specifically, $\mathcal{L}_{adv}(G(\mathcal{X}_s), G(\mathcal{X}_{t\to s}))$ is defined as,

$$\begin{aligned}\mathcal{L}_{adv}(G(\mathcal{X}_s), G(\mathcal{X}_{t\to s})) = \mathbb{E}[logD(G(\mathcal{X}_s))] + \\ \mathbb{E}[log(1 - D(G(\mathcal{X}_{t\to s})))],\end{aligned} \tag{2.2}$$

which enforces G to learn domain-invariant features by confusing the discriminator D. It is noteworthy that we regard the segmentation outputs $G(\mathcal{X}_s)$ and $G(\mathcal{X}_{t\to s})$ as features in our study. This is based on the observation that $\mathcal{X}_s$ and $\mathcal{X}_{t\to s}$ share significant similarities in terms of spatial layouts and structures [19].

2.3.4 Image Reconstruction from the Label Space

To encourage G to generate segmentation outputs that are semantic consistent, we introduce a reconstruction network $\mathcal{M}$ to reconstruct $\mathcal{X}_\phi$ from $G(\mathcal{X}_\phi) \in \mathbb{R}^{H_\phi \times W_\phi \times C}$, where (H_ϕ, W_ϕ) indicates image size, C represents the number of label classes, and the subscript ϕ can be either s or $t \to s$ to denote the source or the target domain. However, directly reconstructing images from the feature space fails to provide semantic consistency constraint to G. On the one hand, $G(\mathcal{X}_\phi)$ encodes rich information which makes the image reconstruction quite straightforward. As illustrated in Figure 2.4, in just a few epochs, the reconstructed images derived from $\mathcal{M}$ are almost identical to the input images. On the other hand, to enforce cross-domain features with the same category close to each other, it is essential to perform the reconstruction based on the label space. Unfortunately, $G(\mathcal{X}_\phi)$ lies in the feature space instead. To overcome these limitations, the most clear-cut way is to convert $G(\mathcal{X}_\phi)$ to have zeros everywhere except where the index of each maximum value in the last dimension. Doing so formulates the categorical representation of the predicted label that corresponds to $G(\mathcal{X}_\phi)$. Nevertheless, such conversion is non-differentiable and cannot be trained using standard backpropagation.

Driven by the softmax action selection which is commonly used in the reinforcement learning, we apply Boltzmann distributed probabilities to approximate the semantic label map of $G(\mathcal{X}_\phi)$, which is defined as,

$$\Omega_\phi{}^{(h,w,i)} = \frac{exp(G(\mathcal{X}_\phi)^{(h,w,i)}/\tau)}{\sum_{j=1}^{c} exp(G(\mathcal{X}_\phi)^{(h,w,j)}/\tau)}, \tag{2.3}$$

where τ is a temperature parameter. This conversion is continuous and differentiable, therefore, we use $\mathcal{M}$ to reconstruct input images $\mathcal{X}_\phi$ from Ω_ϕ (Figure 2.4).

To synthesize high-resolution images from the semantic label map, we use conditional GANs [83] to model the conditional distribution of $\mathcal{X}_\phi$ given Ω_ϕ. To

this end, we introduce $\mathcal{M}$ and multi-scale domain discriminators D_k for $k = 1, 2, 3$. $\mathcal{M}$ is designed to reconstruct $\mathcal{X}_\phi$ from Ω_ϕ, and D_k aims to distinguish $\mathcal{X}_\phi$ from $\mathcal{M}(\Omega_\phi)$. Specifically, $\mathcal{M}$ follows the architecture proposed in [84], while D_k is based on PatchGAN [83] that penalizes structure at the scale of image patches. All D_k follow the same network architecture. Besides $\mathcal{X}_\phi$ and $\mathcal{M}(\Omega_\phi)$ themselves, they are downsampled by a factor of 2 and 4 to obtain pyramid of 3 scales for D_1, D_2, and D_3, respectively. It is worth mentioning that D_k is essential to differentiate real and reconstructed images with high resolution [85], owing to its ability in providing large receptive field. The objective function is given by,

$$\begin{aligned}\mathcal{L}_{adv}^{\phi} = \sum\nolimits_{k=1}^{3} [\mathbb{E}[log D_k(\Omega_\phi, \mathcal{X}_\phi)] + \\ \mathbb{E}[log(1 - D_k(\Omega_\phi, \mathcal{M}(\Omega_\phi)))]]\end{aligned} \tag{2.4}$$

To further enforce semantic consistency between $\mathcal{X}_\phi$ and $\mathcal{M}(\Omega_\phi)$, we introduce a cycle-reconstruction loss $\mathcal{L}_{rec}^{\phi}$ to match their feature representations. $\mathcal{L}_{rec}^{\phi}$ contains a VGG perceptual loss and a discriminator feature matching loss, which is defined as,

$$\begin{aligned}\mathcal{L}_{rec}^{\phi} = \mathbb{E} \sum_{m=1}^{M} [||V^{(m)}(\mathcal{M}(\Omega_\phi)) - V^{(m)}(\mathcal{X}_\phi)||_1] + \\ \mathbb{E} \sum_{k=1}^{3} \sum_{n=1}^{N} [||D_k^{(n)}(\Omega_\phi, \mathcal{X}_\phi)) - D_k^{(n)}(\Omega_\phi, \mathcal{M}(\Omega_\phi))||_1]\end{aligned} \tag{2.5}$$

where V is a VGG19-based model for extracting high-level perceptual information [84], M and N represent the total number of layers in V and D_k for matching intermediate representations. Note that $\mathcal{L}_{rec}^{\phi}$ penalizes Ω_ϕ when it deviates from the corresponding image $\mathcal{X}_\phi$ in terms of semantic consistency. In this way, $\mathcal{M}$ enables to map features from $\mathcal{X}_{t \to s}$ closer to the features from $\mathcal{X}_s$ with the same label.

Taken together, the training objective of our framework is formulated as,

$$\min_{G, \mathcal{M}} \max_{D, D_1, D_2, D_3} \mathcal{L}_G + \alpha(\mathcal{L}_{adv}^{s} + \mathcal{L}_{adv}^{t \to s}) + \beta(\mathcal{L}_{rec}^{s} + \mathcal{L}_{rec}^{t \to s}) \tag{2.6}$$

where α and β leverage the importance of losses above. Notably, our method is able to implicitly encourage G to generate semantic-consistent segmentation labels for the target domain.

2.4 Experiments

In this section, a comprehensive evaluation is performed on two domain adaption tasks to assess our framework for semantic segmentation. Specifically, we consider the large distribution shift of adapting from synthetic (i.e., GTA5 [49] and SYNTHIA [50]) to the real images in Cityscapes [66]. A thorough comparison with the state-of-the-art methods and extensive ablation studies are also carried out to verify the effectiveness of each component in our framework.

2.4.1 Datasets

Cityscapes is one of the benchmarks for urban scene understanding, which is collected from 50 cities with varying scene layouts and weather conditions. The 5,000 finely-annotated images from this dataset are used in our study, which contains 2,975 training images, 500 validation images, and 1,525 test images. Each image with a resolution of 2048 $\times$ 1024. The GTA5 dataset is synthesized from the game Grand Theft Auto V (GTA5), including a total of 24,966 labeled images whose annotations are compatible with Cityscapes. The resolution of each image is 1914 $\times$ 1052. The SYNTHIA-RAND-CITYSCAPES (or SYNTHIA for short) contains 9,400 pixel-level annotated images (1280 $\times$ 760), which are synthesized from a virtual city. Following the same setting reported in the previous studies, we use the labeled SYNTHIA or GTA5 dataset as the source domain, while using the unlabeled training dataset in the CITYSCAPES as the target domain. Only the 500 labeled validation images from CITYSCAPES are used as test data in all of our experiments.

2.4.2 Network Architecture

We use two segmentation baseline models, i.e., FCN-VGG16 and DeepLab-ResNet101 to investigate the effectiveness and generalizability of our framework. Specifically, FCN-VGG16 is the combination of FCN-8s [48] and VGG16 [86], while DeepLab-ResNet101 is obtained by integrating DeepLab-V2 [70] into ResNet101 [7]. These two segmentation models share the same discriminator which has 5 convolution layers with channel number $\{64, 128, 256, 512, 1\}$. For each layer, a leaky ReLU parameterized by 0.2 is followed, except the last one. The kernel size and stride are set to 4×4 and 2, respectively. The reconstruction model follows the architecture in [84], containing 3 convolution layers (kernel 3×3 and stride 1), 9 ResNet blocks (kernel 3×3 and stride 2), and another 3 transposed convolution layers (kernel 3×3 and stride 2) for upsampling. The 3 multi-scale discriminators share the identical network, each of which follows the architecture of PatchGAN [83].

2.4.3 Implementation Details

Our framework is implemented with PyTorch [87] on two TITAN Xp GPUs, each of which with 12GB memory. The batch size is set to one for training all the models discussed above. Limited by the GPU memory space, the translation network is first trained to perform target-to-source image translation by using Adam optimizer [88]. The initial learning rate is set to 0.0001, which is reduced by half after every 100,000 iterations. We use momentum {0.5, 0.999} with weight decay 0.0001. The maximum training iteration is $1000k$.

DeepLab-ResNet101 is trained using Stochastic Gradient Descent optimizer with initial learning rate 2.5×10^{-4}. The polynomial decay with power 0.9 is applied to the learning rate. The momentum and weight decay are set to 0.9 and 5×10^{-4}, respectively. For FCN-VGG16, the Adam optimizer with momentum {0.9, 0.99} and

Table 2.1: A performance comparison of our method with other state-of-the-art models on "GTA5 to Cityscapes". The performance is measured by the intersection-over-union (IoU) for each class and mean IoU (mIoU). Two base architectures, i.e., VGG16 (V) and ResNet101 (R) are used in our study.

GTA5 → Cityscapes																					
	Base	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	**mIoU**
Source only	R	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
SIBAN [56]	R	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
CLAN [57]	R	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
DISE [80]	R	91.5	47.5	82.5	31.3	25.6	33.0	33.7	25.8	82.7	28.8	82.7	62.4	30.8	85.2	27.7	34.5	6.4	25.2	24.4	45.4
IntraDA [36]	R	90.6	37.1	82.6	30.1	19.1	29.5	32.4	20.6	**85.7**	40.5	79.7	58.7	31.1	**86.3**	31.5	48.3	0.0	30.2	35.8	46.3
BDL [34]	R	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	**43.6**	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
CrCDA [35]	R	92.4	**55.3**	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6
SIM [37]	R	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
Kim et al. [38]	R	**92.9**	55.0	**85.3**	34.2	**31.1**	34.9	**40.7**	34.0	85.2	40.1	**87.1**	61.0	31.1	82.5	32.3	42.9	0.3	**36.4**	46.1	50.2
FDA-MBT [39]	R	92.5	53.3	82.4	26.5	27.6	**36.4**	40.6	**38.9**	82.3	39.8	78.0	**62.6**	**34.4**	84.9	34.1	**53.1**	**16.9**	27.7	**46.4**	**50.45**
Ours	R	90.8	41.4	84.7	**35.1**	27.5	31.2	38.0	32.8	85.6	42.1	84.9	59.6	**34.4**	85.0	**42.8**	52.7	3.4	30.9	38.1	49.5
Source only	V	26.0	14.9	65.1	5.5	12.9	8.9	6.0	2.5	70.0	2.9	47.0	24.5	0.0	40.0	12.1	1.5	0.0	0.0	0.0	17.9
SIBAN [56]	V	83.4	13.0	77.8	20.4	17.5	24.6	22.8	9.6	81.3	29.6	77.3	42.7	10.9	76.0	22.8	17.9	5.7	14.2	2.0	34.2
ASN [19]	V	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
CyCADA [18]	V	85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
CLAN [57]	V	88.0	30.6	79.2	23.4	20.5	26.1	23.0	14.8	81.6	34.5	72.0	45.8	7.9	80.5	26.6	29.9	0.0	10.7	0.0	36.6
CrDoCo [62]	V	89.1	33.2	80.1	26.9	25.0	18.3	23.4	12.8	77.0	29.1	72.4	**55.1**	20.2	79.9	22.3	19.5	1.0	20.1	18.7	38.1
CrCDA [35]	V	86.8	37.5	80.4	30.7	18.1	26.8	25.3	15.1	81.5	30.9	72.1	52.8	19.0	82.1	25.4	29.2	10.1	15.8	3.7	39.1
BDL [34]	V	89.2	40.9	81.2	29.1	19.2	14.2	29.0	19.6	83.7	35.9	80.7	54.7	23.3	82.7	25.8	28.0	2.3	25.7	19.9	41.3
FDA-MBT [39]	V	86.1	35.1	80.6	30.8	20.4	27.5	30.0	26.0	82.1	30.3	73.6	52.5	21.7	81.7	24.0	30.5	**29.9**	14.6	24.0	42.2
Kim et al. [38]	V	**92.5**	**54.5**	**83.9**	**34.5**	**25.5**	**31.0**	30.4	18.0	**84.1**	**39.6**	**83.9**	53.6	19.3	81.7	21.1	13.6	17.7	12.3	6.5	42.3
SIM [37]	V	88.1	35.8	83.1	25.8	23.9	29.2	28.8	**28.6**	83.0	36.7	82.3	53.7	22.8	82.3	26.4	**38.6**	0.0	19.6	17.1	42.4
Ours	V	90.1	41.2	82.2	30.3	21.3	18.3	**33.5**	23.0	**84.1**	37.5	81.4	54.2	**24.3**	**83.0**	**27.6**	32.0	8.1	**29.7**	**26.9**	**43.6**

initial learning rate 1×10^{-5} is used for training. The learning rate is decreased using step decay with step size 50000 and drop factor 0.1. In equation 2.1, λ is set to 1×10^{-3} for DeepLab-ResNet101 and 1×10^{-4} for FCN-VGG16.

The reconstruction network is first pre-trained by reconstructing source images $\mathcal{X}_s$ from the corresponding labels Y_s. We use the Adam optimizer with initial learning rate 2×10^{-4} and momentum $\{0.5, 0.999\}$, where the learning rate is linearly decreased to zero. In equation 2.6, we set $\beta = 10$. α is set to 0.01 and 0.001 for FCN-VGG16 and DeepLab-ResNet101 respectively.

Table 2.2: A performance comparison of our method with other state-of-the-art models on "SYNTHIA to Cityscapes". The performance is measured by the IoU for each class and mIoU. Two base architectures, i.e., VGG16 (V) and ResNet101 (R) are used in our study.

SYNTHIA → Cityscapes																		
	Base	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	bus	motorbike	bicycle	**mIoU**
Source only	R	55.6	23.8	74.6	—	—	—	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6
ASN [19]	R	84.3	42.7	77.5	—	—	—	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7
DISE [80]	R	91.7	**53.5**	77.1	—	—	—	6.2	7.6	78.4	81.2	55.8	19.2	82.3	30.3	17.1	34.3	48.8
IntraDA [36]	R	84.3	37.7	79.5	—	—	—	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	48.9
Kim et al. [38]	R	**92.6**	53.2	79.2	—	—	—	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	49.3
DADA [54]	R	89.2	44.8	**81.4**	—	—	—	8.6	11.1	**81.8**	84.0	54.7	19.3	79.7	40.7	14.0	38.8	49.8
CrCDA [35]	R	86.2	44.9	79.5	—	—	—	9.4	11.8	78.6	**86.5**	57.2	26.1	76.8	39.9	21.5	32.1	50.0
BDL [34]	R	86.0	46.7	80.3	—	—	—	14.1	11.6	79.2	81.3	54.1	27.9	73.7	**42.2**	25.7	45.3	51.4
SIM [37]	R	83.0	44.0	80.3	—	—	—	17.1	15.8	80.5	81.8	59.9	**33.1**	70.2	37.3	28.5	45.8	52.1
FDA-MBT [39]	R	79.3	35.0	73.2	—	—	—	**19.9**	**24.0**	61.7	82.6	**61.4**	31.1	**83.9**	40.8	**38.4**	**51.1**	52.5
Ours	R	85.1	44.5	81.0	—	—	—	16.4	15.2	80.1	84.8	59.4	31.9	73.2	41.0	32.6	44.7	**53.1**
CrCDA [35]	V	74.5	30.5	78.6	6.6	**0.7**	21.2	2.3	8.4	77.4	79.1	45.9	16.5	73.1	24.1	9.6	14.2	35.2
ROAD-Net [89]	V	77.7	30.0	77.5	9.6	0.3	25.8	10.3	15.6	77.6	79.8	44.5	16.6	67.8	14.5	7.0	23.8	36.2
SPIGAN [90]	V	71.1	29.8	71.4	3.7	0.3	**33.2**	6.4	15.6	81.2	78.9	52.7	13.1	75.9	25.5	10.0	20.5	36.8
GIO-Ada [53]	V	78.3	29.2	76.9	**11.4**	0.3	26.5	10.8	17.2	81.7	**81.9**	45.8	15.4	68.0	15.9	7.5	30.4	37.3
TGCF-DA [91]	V	**90.1**	**48.6**	**80.7**	2.2	0.2	27.2	3.2	14.3	**82.1**	78.4	54.4	16.4	**82.5**	12.3	1.7	21.8	38.5
BDL [34]	V	72.0	30.3	74.5	0.1	0.3	24.6	10.2	25.2	80.5	80.0	54.7	23.2	72.7	24.0	7.5	44.9	39.0
FDA-MBT [39]	V	84.2	35.1	78.0	6.1	0.44	27.0	8.5	22.1	77.2	79.6	55.5	19.9	74.8	24.9	**14.3**	40.7	40.5
Ours	V	73.7	29.6	77.6	1.0	0.4	26.0	**14.7**	**26.6**	80.6	81.8	**57.2**	**24.5**	76.1	**27.6**	13.6	**46.6**	**41.1**

2.4.4 GTA5 → Cityscapes

We carry out the adaptation from GTA5 to Cityscapes by following the same evaluation protocol as previously reported in [19, 34]. The overall quantitative performance is assessed on 19 common classes (e.g., road, wall, and car) between these two domains. As shown in Table 2.1, we demonstrate competitive performance against ResNet101-based methods, but are inferior to two newly published models [38, 39]. For the VGG16-based backbone, however, we are able to achieve the best results compared to existing state-of-the-art methods including [38, 39]. Specifically, our method surpasses the source-only model (without adaptation) by 12.9% and 25.7% on ResNet101 and VGG16, respectively. Compared with CyCADA [18] and BDL [34] that rely on source-to-target translation, we demonstrate significant improvements (i.e., 8.2% and 2.3% on VGG16) by reducing image translation bias. CLAN [57] aims to enforce local semantic consistency by a category-level adversarial network. However,

Table 2.3: Ablation study of the target-to-source translation and the reconstruction network. S → T and T → S indicate source-to-target and target-to-source translation.

Base	S → T	T → S	Reconstruction	GTA5	SYNTHIA
R	✓			48.5	51.4
R		✓		49.1	52.0
R		✓	✓	49.5	53.1
V	✓			41.3	39.0
V		✓		42.3	40.1
V		✓	✓	43.6	41.1

such a strategy fails to account for the global semantic consistency. Our reconstruction network shares a similar spirit with CLAN in terms of joint distribution alignment but enables us to enforce semantic consistency from a global view. As a consequence, we get 6.3% and 7.0% improvement on ResNet101 and VGG16, respectively.

2.4.5 SYNTHIA → Cityscapes

We then evaluate our framework on the adaptation from SYNTHIA to Cityscapes based on 13 classes on ResNet101 and 16 classes on VGG16. The results exhibit that our method outperforms other competing methods on average as shown in Table 2.2. Both ASN [19] and BDL [34] adapt output space in their models. However, simply aligning segmentation outputs may lead to negative transfer issue, owing to the dramatic differences of the layout and structure between SYNTHIA and Cityscapes. We achieve 6.4% and 1.7% improvement than ASN and BDL on ResNet101, respectively. It is noteworthy that we also outperform [39] on both ResNet101 and VGG16-based backbone.

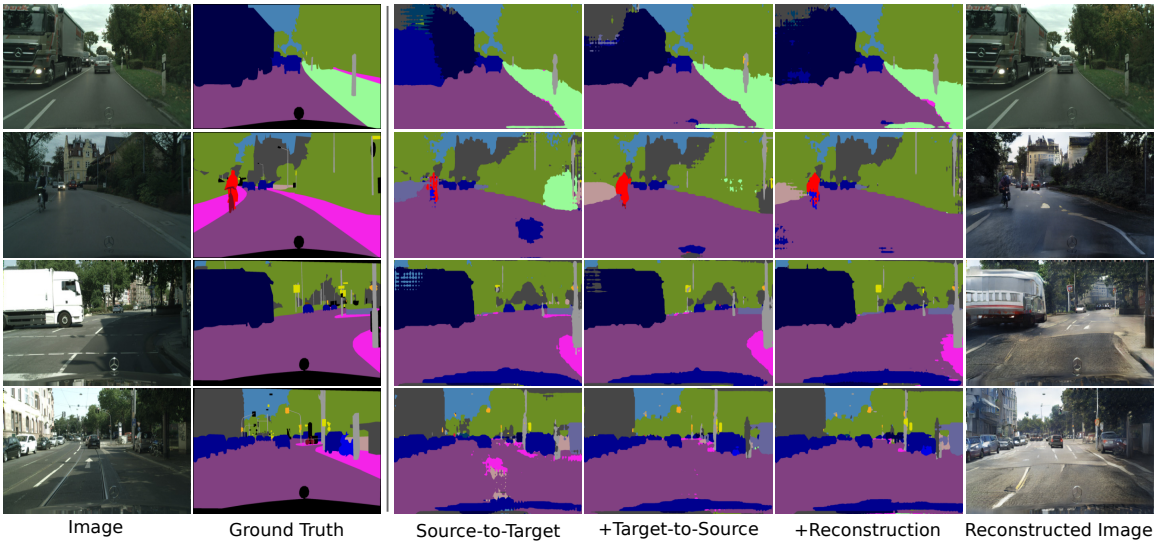

Figure 2.5: Qualitative examples of semantic segmentation results in Cityscapes. For each target-domain image (first column), its ground truth and the corresponding segmentation prediction from the baseline model (source-to-target translation) are given. The following are predictions of our method by incorporating target-to-source translation and reconstruction, together with the reconstructed image.

2.4.6 Ablation Study

2.4.6.1 Target-to-source Translation and Reconstruction

For GTA5 to Cityscapes, 0.6% improvement is achieved by considering target-to-source translation on ResNet101 compared to the source-to-target translation model (Table 2.3). By further enforce semantic consistency through a reconstruction network, our method achieves 49.5 mIoU. Similar improvements are also observed on VGG16, with 1.0% improvement by performing target-to-source translation. The prediction power of our method is further boosted by combining translation and reconstruction, giving rise to another 1.3% mIoU improvement. The qualitative study of each module in our method is showcased in Figure 2.5.

For SYNTHIA to Cityscapes, we achieve a performance boost of 0.6% and 1.1% by considering target-to-source translation on ResNet101 and VGG16, respectively

Table 2.4: Ablation study of the temperature τ on GTA5 $\rightarrow$ Cityscapes.

τ	0.0001	0.001	0.01	0.1	1
mIoU	42.7	**43.6**	42.8	42.9	41.5

Table 2.5: Ablation study of the feature space vs. label space reconstruction.

	Feature space	Label space
GTA5 $\rightarrow$ Cityscapes	41.48	43.6
SYNTHIA $\rightarrow$ Cityscapes	40.13	41.1

(Table 2.3). The performance gain is 1.1% and 1.0% by incorporating the reconstruction network. Our results prove the effectiveness of target-to-source translation and reconstruction in adapting domain knowledge for semantic segmentation.

2.4.6.2 Parameter Analysis

We investigate the sensitivity of temperature parameter τ in this section and find that $\tau = 0.001$ achieves the best performance (Table 2.4). Therefore, τ is set to 0.001 in all of our experiments to approximate semantic label maps.

2.4.6.3 Feature Space VS. Label Space Reconstruction

We also evaluate the feature space reconstruction based on the VGG16-based backbone. Table 2.5 highlights the benefits of our label-driven reconstruction that enforces semantic consistency of target images and their predicted labels.

2.4.6.4 Reconstruction loss

Table 2.6 shows the complementary role of VGG perceptual loss and discriminator feature matching loss (equation 2.5) in maintaining semantic consistency.

Table 2.6: Ablation study of the reconstruction loss on GTA5 → Cityscapes with VGG16 backbone.

VGG	Discriminator	mIoU
		41.53
✓		42.82
	✓	41.95
✓	✓	43.6

2.5 Summary and Discussion

In this chapter, we propose a novel framework that exploits cross-domain adaptation in the context of semantic segmentation. Specifically, we translate images from the target domain to the source domain to reduce image translation bias and the computational cost. To enforce cross-domain features with the same category close to each other, we reconstruct both source and target images directly from the label space. Experiments demonstrate that our method achieves significant improvement in adapting from GTA5 and SYNTHIA to Cityscapes.

As discussed in Chapter 1, there are two primary issues in transfer learning, i.e., what and how to transfer domain knowledge across two domains. Existing methods mainly focus on adapting domain-invariant features (what to transfer) through adversarial learning (how to transfer). Context dependency is essential for semantic segmentation, however, its transferability is still not well understood. Furthermore, how to transfer contextual information across two domains remains unexplored. Therefore, a promising research topic would be to incorporating contextual information into UDA. This will be our main focus in the next chapter.

CHAPTER 3

Context-Aware Domain Adaptation in Semantic Segmentation

In this chapter, we propose a cross-attention mechanism based on self-attention to capture context dependencies between two domains and adapt transferable context. To achieve this goal, we design two cross-domain attention modules to adapt context dependencies from both spatial and channel views. Specifically, the spatial attention module captures local feature dependencies between each position in the source and target image. The channel attention module models semantic dependencies between each pair of cross-domain channel maps. To adapt context dependencies, we further selectively aggregate the context information from two domains. The superiority of our method over existing state-of-the-art methods is empirically proved on "GTA5 to Cityscapes" and "SYNTHIA to Cityscapes".

3.1 Introduction

Semantic segmentation aims to predict pixel-level labels for the given images [48, 70], which has been widely recognized as one of the fundamental tasks in computer vision. Unfortunately, the manual pixel-wise annotation for large-scale segmentation datasets is extremely time-consuming and requires massive amounts of labor efforts. As a tradeoff, synthetic datasets [49, 50] with freely-available labels offer a promising alternative by providing considerable data for model training. However, the domain discrepancy between synthetic (source) and real (target) images is still the central challenge to effectively transfer knowledge across domains. To overcome this limitation, the key idea of existing methods is to leverage knowledge from a source domain to

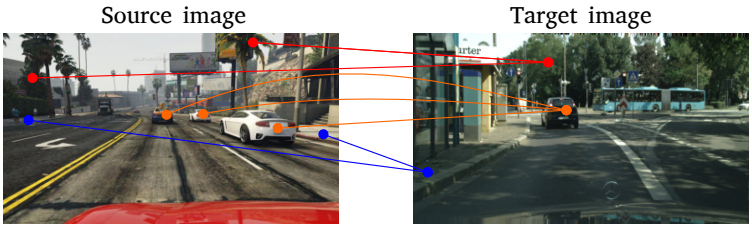

Figure 3.1: An example of cross-domain context. The source and target images share similar context information at the spatial and semantic level. The red line, orange line, and blue line denote vegetation, car, and sidewalk across two domains, respectively.

enhance the learning performance of a target domain. Such a strategy is mainly inspired by the recent advances in unsupervised domain adaptation for image classification [8].

Conventional domain adaptation methods in image classification attempt to learn domain-invariant feature representations by directly minimizing the representation distance between two domains [12, 13, 28], encouraging a common feature space through an adversarial objective [74, 25], or automatically determining what and where to transfer via meta-learning [76, 92]. Motivated by this, various domain adaptation methods for semantic segmentation are proposed recently. Among them, the most common practices are based on feature alignment [17, 32], structure adaptation [19, 89], adversarial learning [93, 90, 94, 95], curriculum adaptation [51, 52], self training [55, 34, 46, 36], and image-to-image translation [18, 59, 34, 62, 53, 46]. Despite remarkable performance improvement achieved by these methods, they fail to explicitly consider the contextual dependencies across the source and target domains which is essential for scene understanding [96, 97]. As illustrated in Figure 3.1, the source and target images share a much similar semantic context such as vegetation, car, and sidewalk, although their appearances (*e.g.*, scale, texture, and illumination) are quite different. However, how to adapt context information across two domains remains unexplored.

Inspired by this, we propose a novel domain adaptation framework named cross-domain attention network (CDANet), designed for urban-scene semantic segmentation. The key idea of CDANet is to leverage cross-domain context dependencies from both a local and global perspective. To achieve this goal, we innovatively design a cross-attention mechanism which contains two cross-domain attention modules to capture mutual context dependencies between source and target domains. Given that same objects with different appearances and scales often share similar features, we introduce a cross-domain spatial attention module (CD-SAM) to capture local feature dependencies between any two positions in a source image and a target image. The CD-SAM involves two directions (*i.e.*, "source-to-target" and "target-to-source") to adaptively aggregate cross-domain features to learn common context information. On the forward direction (or "source-to-target"), CD-SAM updates the feature at each position in the source image as the weighted sum of features at all positions in the target image. The weights are computed based on the similarity of source and target features at each position. Similarly, the backward direction (or "target-to-source") updates the target feature at each position based on the attention to features at all positions in the source image. In consequence, spatial contexts from the source domain are encoded in the target domain, and vice versa. To model the associations between different semantic responses across two domains, we introduce a cross-domain channel attention module (CD-CAM) which has the same bidirectional structure as CD-SAM. The CD-CAM is designed for contextual information aggregation through capturing the channel feature dependencies between any two channel maps in the source and target image. In such a way, common semantic contexts are shared by both domains. CD-SAM and CD-CAM play a complementary role for context adaptation and their outputs are further merged to provide better feature representations for scene understanding.

Our main contributions are summarized as follows: (i) We propose a novel cross-attention mechanism that enables to transfer of context dependencies across two domains. This is the first-of-its-kind study that investigates the transferability of context information in the domain adaptation; (ii) Two cross-domain attention modules are proposed to capture and adapt context dependencies at both spatial and channel levels. This allows us to learn the common semantic context shared by source and target domains; and (iii) Comprehensive empirical studies demonstrate the superiority of our method over the existing state of the art on two benchmark settings, *i.e.*, "GTA5 to Cityscapes" and "SYNTHIA to Cityscapes".

3.2 Related Work

3.2.1 Domain Adaptation for Semantic Segmentation

Inspired by the Generative Adversarial Network [30], Hoffman *et al.*[17] propose the first domain adaptation model for semantic segmentation by learning domain-invariant features through adversarial training. To rule out task-independent factors during feature alignment, SIBAN [56] purifies significance-aware features before the adversarial adaptation to facilitate feature adaptation and stabilize the adversarial training. However, these global adversarial methods ignore to align the category-level joint distribution, which may disturb well-aligned features. To alleviate this problem, Luo *et al.*propose a category-level adversarial network to encourage local semantic consistency through reweighting the adversarial loss for each feature [57]. Similarly, [98] proposes a fine-grained adversarial learning strategy for class-level feature alignment. Based on the hypothesis that structure information plays an essential role in semantic segmentation, Chang *et al.*adapt structure information by learning domain-invariant structure [80]. This is achieved by disentangling the domain-

invariant structure of a given image from its domain-specific texture information. AdaptSetNet moves forward by further considering structured output adaptation which is based on the observation that segmentation outputs of the source and target domains share substantial similarities [19]. Different from AdaptSetNet, we apply three domain discriminators to perform output adaptation on the segmentation outputs from CD-SAM, CD-CAM, and the aggregation of these two modules.

Most recently, image-to-image translation [63] has proved its effectiveness in domain adaptation [18, 60, 62]. The key idea is to translate images from the source domain to the target domain by using an image translation model and use the translated images for adapting cross-domain knowledge through a segmentation adaptation model. Rather than keeping the image translation model unchanged after obtaining translated images, BDL [34] applies a bidirectional learning framework to alternatively optimize the image translation model and the segmentation model. Similar to [55, 36], a self-supervised learning strategy is also used in BDL to generate pseudo labels for target images and re-training the segmentation model with these labels. Although BDL achieves the new state of the art, it is limited in its ability to consider the cross-domain context dependencies. To overcome this limitation, we introduce two cross-domain attention modules to adapt context information between source and target domains.

3.2.2 Context-Aware Embedding

It has been long known that context information plays an important role in perceptual tasks such as semantic segmentation [99]. Zhang *et al.*[96] propose a context encoding module to capture the semantic context of scenes and selectively emphasize or de-emphasize class-dependent feature maps. To aggregate image-adapted context, MSCI [100] further considers multi-scale context embedding and spatial

relationships among super-pixels in a given image. Following the success of attention mechanism [101] in image generation [102] and sentence embedding [103], recent studies have highlighted the potential of self-attention in capturing context dependencies [104, 97]. Specifically, Zhao *et al.*[97] introduce a point-wise spatial attention network to aggregate long-range contextual information. Their model mainly draws its strength from the self-adaptively predicted attention maps which can take full advantage of both nearby and distant information of each pixel. DANet [104] adaptively integrates local features with their global dependencies through a position attention module and a channel attention module. These two modules are considered to be able to capture spatial and semantic interdependencies, and in turn, facilitate scene understanding. Similarly, CBAM [105] sequentially infers attention maps along the channel and spatial dimensions in order to adaptively refine the intermediate features. As opposed to capturing contextual information within a single domain as previously reported, we design an innovative cross-attention mechanism to model context dependencies between two different domains, which is essential for context adaptation.

3.3 Methodology

In this section, we begin by briefing the key idea of our framework. We then detail the proposed cross-attention mechanism which contains two cross-domain attention modules for adapting context dependencies between a source and a target domain.

3.3.1 Overview

Given a set of source-domain images $\mathcal{X}_s$ with pixel-wise labels Y_s and a set of target-domain images $\mathcal{X}_t$ without any annotation. Our goal is to train a segmentation model that can provide accurate prediction to $\mathcal{X}_t$. To achieve this, $\mathcal{X}_s$ is first translated from the source domain to the target domain using CycleGAN [63]. The translated

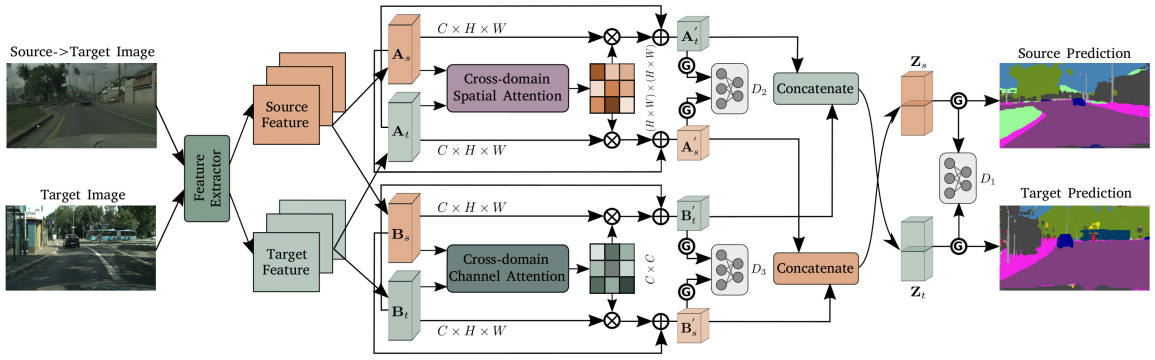

Figure 3.2: An overview of the proposed framework. It applies a feature extractor (*i.e.*, ResNet101 or VGG16) to learn source and target features. Two cross-domain attention modules (*i.e.*, CD-SAM and CD-CAM) are designed to adapt spatial and semantic context information across source and target domains. A classifier G is used to predict segmentation output based on the features from CD-SAM and CD-CAM. Our framework contains three discriminators (*i.e.*, D_1, D_2, and D_3) for output adaptation by enforcing the source output be indistinguishable from the target output.

images $\mathcal{X}_s^{'} = \mathcal{F}(\mathcal{X}_s)$ (where $\mathcal{F}$ denotes the image translation model) share the same semantic labels with $\mathcal{X}_s$ but with common visual appearance as $\mathcal{X}_t$. Motivated by the self-training strategy, we follow the same idea in [34, 36] to generate pseudo labels Y_t^{st} for $\mathcal{X}_t$ with high prediction confidence. Coordinated with these translated images and pseudo labels, we introduce a cross-attention mechanism for domain adaptation of semantic segmentation by leveraging cross-domain contextual information (Figure 3.2). First, a feature extractor E is applied to get source feature $E(\mathcal{X}_s^{'})$ and target feature $E(\mathcal{X}_t)$ which are 1/8 of the corresponding input image size. Then a linear interpolation is applied to $E(\mathcal{X}_s^{'})$ and $E(\mathcal{X}_t)$ to match their spatial size. After that, two parallel convolution layers are applied to $E(\mathcal{X}_s^{'})$ and $E(\mathcal{X}_s)$ to generate feature pairs $\{\mathbf{A}_s, \mathbf{A}_t\}$ and $\{\mathbf{B}_s, \mathbf{B}_t\}$, respectively. $\{\mathbf{A}_s, \mathbf{A}_t\}$ is then fed into CD-SAM to adapt spatial-level context, while CD-CAM adapts channel-level context based on $\{\mathbf{B}_s, \mathbf{B}_t\}$.

For each module, two directions, *i.e.*, forward direction ("source-to-target") and backward direction ("target-to-source") are involved. Take the CD-SAM as an

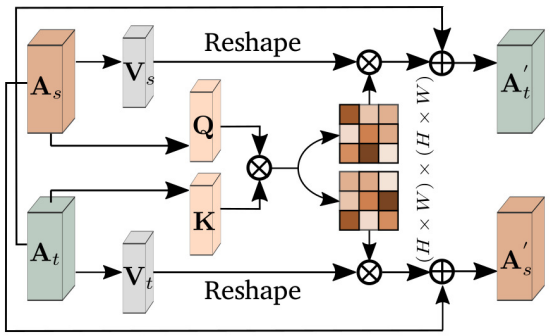

Figure 3.3: Cross-domain spatial attention module.

example, an energy map is first obtained based on $\{\mathbf{A}_s, \mathbf{A}_t\}$. This energy map is further divided into two attention matrices denoted by $\Gamma_{s\to t}$ and $\Gamma_{t\to s}$. During the forward direction, we perform a matrix multiplication between target features and $\Gamma_{s\to t}$. The result is then summed with the original source features in an element-wise manner. For the backward direction, a matrix multiplication is conducted between source features and $\Gamma_{t\to s}$. After that, an element-wise summation between the obtained results and original target features is carried out. The CD-CAM follows the same setting above except that the energy map is calculated in the channel dimension. The final source feature and target feature are obtained by aggregating the outputs from these two attention modules, which are then fed into a classifier G for semantic segmentation.

3.3.2 Cross-Domain Spatial Attention Module

The goal of CD-SAM is to adapt spatial contextual information across two domains. To achieve this, we introduce the forward direction ("source-to-target") to augment source features by selectively aggregating target features based on their similarities. We further introduce the backward direction ("target-to-source") to update target features by aggregating source features in the same way.

The architecture of CD-SAM is illustrated in Figure 3.3. Given $\mathbf{A}_s \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{A}_t \in \mathbb{R}^{C\times H\times W}$ (C denotes the channel number and $H \times W$ indicates the spatial size), two parallel convolution layers are applied to generate $\mathbf{Q} \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{K} \in \mathbb{R}^{C\times H\times W}$, respectively. $\mathbf{A}_s$ and $\mathbf{A}_t$ are also fed into another convolution layer to obtain $\mathbf{V}_s \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{V}_t \in \mathbb{R}^{C\times H\times W}$. We reshape $\mathbf{Q}$, $\mathbf{V}_s$, $\mathbf{K}$, and $\mathbf{V}_t$ to $C \times N$, where $N = H \times W$. To determine spatial context relationships between each position in $\mathbf{A}_s$ and $\mathbf{A}_t$, an energy map $\Phi \in \mathbb{R}^{N\times N}$ is formulated as $\Phi = \mathbf{Q}^T\mathbf{K}$, where $\Phi^{(i,j)}$ measure the similarity between i^{th} position in $\mathbf{A}_s$ and j^{th} position in $\mathbf{A}_t$. To augment $\mathbf{A}_s$ with spatial context information from $\mathbf{A}_t$ and vice versa, a bidirectional feature adaptation is defined as follows.

During the forward direction, we first define the "source-to-target" spatial attention map as,

$$\Gamma_{s\to t}^{(i,j)} = \frac{exp(\Phi^{(i,j)})}{\sum_{j=1}^{N_t} exp(\Phi^{(i,j)})}, \tag{3.1}$$

where $\Gamma_{s\to t}^{(i,j)}$ indicates the impact of i^{th} position in $\mathbf{A}_s$ to j^{th} position in $\mathbf{A}_t$. To capture spatial context in the target domain, we update $\mathbf{A}_s$ as,

$$\mathbf{A}_s^{'} = \mathbf{A}_s + \lambda_s \mathbf{V}_t \Gamma_{s\to t}^{T}, \tag{3.2}$$

where λ_s leverages the importance of target-domain context and original source features. In this regime, each position in $\mathbf{A}_s^{'}$ has a global context view of target features.

For the backward direction, the "target-to-source" spatial attention map is formulated as,

$$\Gamma_{t\to s}^{(i,j)} = \frac{exp(\Phi^{(i,j)})}{\sum_{i=1}^{N_s} exp(\Phi^{(i,j)})}, \tag{3.3}$$

where $\Gamma_{t\to s}^{(i,j)}$ indicates to what extent the j^{th} position in $\mathbf{A}_t$ attends to the i^{th} position in $\mathbf{A}_s$. Similarly, $\mathbf{A}_t$ is updated by,

$$\mathbf{A}_t^{'} = \mathbf{A}_t + \lambda_t \mathbf{V}_s \Gamma_{t\to s}, \tag{3.4}$$

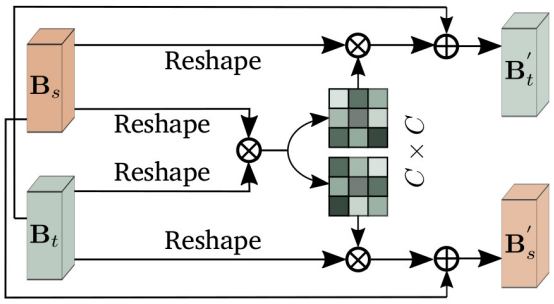

Figure 3.4: Cross-domain channel attention module.

where λ_t leverages the importance of source-domain context and original target features. As a consequence, each position in $\mathbf{A}_s^{'}$ and $\mathbf{A}_t^{'}$ is a combination of their original feature and the weighed sum of features from the opposite domain. Therefore, $\mathbf{A}_s^{'}$ and $\mathbf{A}_t^{'}$ allow us to encode the spatial context of both source and target domains.

3.3.3 Cross-Domain Channel Attention Module

Given $\mathbf{B}_s \in \mathbb{R}^{C \times H \times W}$ and $\mathbf{B}_t \in \mathbb{R}^{C \times H \times W}$, the CD-CAM is designed to adapt semantic context between source and target domains (Figure 3.4) by following the same bidirectional structure as CD-SAM. Different from CD-SAM that applies convolution layers to obtain $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}_s$, and $\mathbf{V}_t$ before measuring spatial relationships. Here, $\mathbf{B}_s$ and $\mathbf{B}_t$ are directly used to capture their semantical context relationships, which allows us to maintain interdependencies between channel maps [104]. Specifically, we reshape both $\mathbf{B}_s$ and $\mathbf{B}_t$ to $C \times N$, where $N = H \times W$. The energy map is defined as $\Theta = \mathbf{B}_t \mathbf{B}_s^T \in \mathbb{R}^{C \times C}$, where $\Theta^{(i,j)}$ denotes the similarity between i^{th} channel in $\mathbf{B}_s$ and j^{th} channel in $\mathbf{B}_t$.

For the forward direction, the "source-to-target" attention map is given by,

$$\Psi_{s \to t}^{(i,j)} = \frac{exp(\Theta^{(i,j)})}{\sum_{j=1}^{C} exp(\Theta^{(i,j)})}, \tag{3.5}$$

where $\Psi_{s\to t}^{(i,j)}$ measures the impact of i^{th} channel in $\mathbf{B}_s$ to j^{th} channel in $\mathbf{B}_t$. To model the cross-domain semantic context dependencies, $\mathbf{B}_s$ is updated by,

$$\mathbf{B}_s^{'} = \mathbf{B}_s + \xi_s \Psi_{s\to t} \mathbf{B}_t, \tag{3.6}$$

where ξ_s leverages the associations between target-domain semantic information and original source features. As a consequence, each channel in $\mathbf{B}_s^{'}$ is augmented by selectively aggregating semantic information from $\mathbf{B}_t$.

During the backward direction, the "target-to-source" attention map is,

$$\Psi_{t\to s}^{(i,j)} = \frac{exp(\Theta^{(i,j)})}{\sum_{i=1}^{C} exp(\Theta^{(i,j)})} \tag{3.7}$$

To take semantic context in $\mathbf{B}_s$ into consideration, we have

$$\mathbf{B}_t^{'} = \mathbf{B}_t + \xi_t \Psi_{t\to s}^{T} \mathbf{B}_s, \tag{3.8}$$

where ξ_t leverages the associations between original target features and semantic contexts from the source domain. It is noteworthy that by considering cross-domain semantic context, our framework is able to further reduce domain discrepancy from the context perspective.

3.3.4 Aggregation of Spatial and Channel Context

To take full advantage of spatial and channel context information, we aggregate the outputs from these two cross-domain attention modules. Specifically, $\mathbf{A}_s^{'}$ and $\mathbf{B}_s^{'}$ are concatenated and then fed into a convolution layer to generate the enhanced source feature $\mathbf{Z}_s \in \mathbb{R}^{C\times H\times W}$. Obviously, $\mathbf{Z}_s$ is enriched by spatial and semantic context dependencies from both source and target domains. The same operation is also performed on $\mathbf{A}_t^{'}$ and $\mathbf{B}_t^{'}$ to obtain $\mathbf{Z}_t \in \mathbb{R}^{C\times H\times W}$.

3.3.5 Training Objective

Our framework contains a segmentation loss $\mathcal{L}_{seg}$ and an adversarial loss $\mathcal{L}_{adv}$. We first feed $\mathbf{Z}_s$ and $\mathbf{Z}_t$ into the classifier G to predict their segmentation outputs $G(\mathbf{Z}_s)$ and $G(\mathbf{Z}_t)$. The segmentation loss of $G(\mathbf{Z}_s)$ is defined as:

$$\mathcal{L}_{seg}(G(\mathbf{Z}_s), Y_s) = -\sum_{i=1}^{H\times W}\sum_{j=1}^{L} Y_s^{(i,j)} G(\mathbf{Z}_s)^{(i,j)}, \tag{3.9}$$

where L is the number of label classes. $\mathcal{L}_{seg}(G(\mathbf{Z}_t), Y_s^{st})$ is defined in a similar way. To adapt structured output space [19], a discriminator D_1 is applied to $G(\mathbf{Z}_s)$ and $G(\mathbf{Z}_t)$ to make them be indistinguishable from each other. To achieve this, an adversarial loss $\mathcal{L}_{adv}(G(\mathbf{Z}_s), G(\mathbf{Z}_t))$ is formulated as,

$$\begin{aligned}\mathcal{L}_{adv}(G(\mathbf{Z}_s), G(\mathbf{Z}_t), D_1) = \ & \mathbb{E}[logD_1(G(\mathbf{Z}_s))]+ \\ & \mathbb{E}[log(1 - D_1(G(\mathbf{Z}_t)))]\end{aligned} \tag{3.10}$$

To encourage $\mathbf{A}_s^{'}$, $\mathbf{A}_t^{'}$, $\mathbf{B}_s^{'}$ and $\mathbf{B}_t^{'}$ to encode useful information for semantic segmentation, they are also fed into the classifier G to predict their segmentation outputs. The overall segmentation loss is given by,

$$\begin{aligned}\mathcal{L}_{seg} = & \mathcal{L}_{seg}(G(\mathbf{Z}_s), Y_s) + \mathcal{L}_{seg}(G(\mathbf{Z}_t), Y_t^{st})+ \\ & \mathcal{L}_{seg}(G(\mathbf{A}_s^{'}), Y_s) + \mathcal{L}_{seg}(G(\mathbf{A}_t^{'}), Y_t^{st})+ \\ & \mathcal{L}_{seg}(G(\mathbf{B}_s^{'}), Y_s) + \mathcal{L}_{seg}(G(\mathbf{B}_t^{'}), Y_t^{st})\end{aligned} \tag{3.11}$$

We also encourage $G(\mathbf{A}_s^{'})$ and $G(\mathbf{A}_t^{'})$ to have similar structured layout, and enforce $G(\mathbf{B}_s^{'})$ to be indistinguishable from $G(\mathbf{B}_t^{'})$. Therefore, the overall adversarial loss can be written as,

$$\begin{aligned}\mathcal{L}_{adv} = & \mathcal{L}_{adv}(G(\mathbf{Z}_s), G(\mathbf{Z}_t), D_1)+ \\ & \mathcal{L}_{adv}(G(\mathbf{A}_s^{'}), G(\mathbf{A}_t^{'}), D_2)+ \\ & \mathcal{L}_{adv}(G(\mathbf{B}_s^{'}), G(\mathbf{B}_t^{'}), D_3),\end{aligned} \tag{3.12}$$

where D_2 and D_3 are two discriminators. Specifically, D_2 aims to discriminate between $G(\mathbf{A}_s^{'})$ and $G(\mathbf{A}_t^{'})$, while D_3 attempts to distinguish between $G(\mathbf{B}_s^{'})$ and $G(\mathbf{B}_t^{'})$.

Taken them together, the training objective of our framework is:

$$\min_{E,G} \max_{D_1,D_2,D_3} \mathcal{L}_{seg} + \lambda \mathcal{L}_{adv} \tag{3.13}$$

where λ controls the importance of $\mathcal{L}_{seg}$ and $\mathcal{L}_{adv}$.

3.4 Experiments

In this section, we evaluate our method on synthetic-to-real domain adaptation for urban scene understanding problem. Extensive empirical experiments and ablation studies are performed to demonstrate out method's superiority over existing state-of-the-art models. We also visualize the cross-domain attention maps to reveal context dependencies between source and target domains.

3.4.1 Datasets

Two synthetic datasets, *i.e.*, GTA5 [49] and SYNTHIA-RAND-CITYSCAPES [50] are used as the source domain in our study, while the Cityscapes [66] is served as the target domain. Specifically, the GTA5 is collected from a photorealistic open-world game known as Grand Theft Auto V, which contains 24,966 images with pixel-accurate semantic labels. The resolution of each image is 1914 $\times$ 1052. SYNTHIA-RAND-CITYSCAPES contains 9,400 images (1280 $\times$ 760) with precise pixel-level semantic annotations, which are generated from a virtual city. Cityscapes is a large-scale street scene datasets collected from 50 cities, including 5,000 images with high-quality pixel-level annotations. These images are split into training (2,975 images), validation (500 images), and test (1,525 images) set, each of which with the resolution of 2048 $\times$ 1024. Following the same setting as previous studies, only the training set from

Cityscapes is used as the target domain, and the validation set is used for performance evaluation.

3.4.2 Implementation Details

3.4.2.1 Network Architecture

The same CycleGAN architecture [63] as reported in BDL [34] is used to translate images from the source domain to the target domain. DeepLab-VGG16 and DeepLab-ResNet101, which are pre-trained on ImageNet [106], are used as our segmentation network by following the same setting in [19]. Both of them use DeepLab-v2 [70] as classifier, while DeepLab-VGG16 uses VGG16 [86] and DeepLab-ResNet101 uses ResNet101 [7] as the feature extractor. The three discriminators used for structured output adaptation have the identical architecture, each of which has 5 convolution layers with kernel 4×4 and stride of 2. The channel number of each layer is {64, 128, 256, 512, 1}. Each layer is followed by a leaky ReLU [107] parameterized by 0.2 except the last one. The CD-SAM contains 3 convolution layers with kernel 1×1 and stride of 1 to obtain the query and key-value pairs. The channel number of these convolution layers are {128, 128, 1024} and {256, 256, 2048} for DeepLab-VGG16 and DeepLab-ResNet101, respectively.

3.4.2.2 Network Training

To train the CycleGAN network, we follow the same setting in BDL [34]. DeepLab-VGG16 is trained using Adam optimizer with initial learning rate 1e-5 and momentum (0.9, 0.99). We apply step decay to the learning rate with step size 50000 and drop factor 0.1. Both DeepLab-ResNet101 and CD-SAM use Stochastic Gradient Descent (SGD) optimizer with momentum 0.9 and weight decay 5e-4. The initial

Table 3.1: The performance comparison by adapting from GTA5 to Cityscapes. Two base architectures (i.e., VGG16 and ResNet101) are used in our study. The comparison is performed on 19 common classes between source and target domains. We use per-class IoU and mean IoU (mIoU) for the performance measurement. The best result in each column is highlighted in bold.

GTA5 to Cityscapes																					
	Architecture	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	**mIoU**
FCNs wild [17]	VGG16	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
CDA [51]		74.9	22.0	71.4	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	14.6	28.9
AdaptSegNet [19]		87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
CyCADA [18]		85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
LSD [81]		88.0	30.5	78.6	25.2	23.5	16.7	23.5	11.6	78.7	27.2	71.9	51.3	19.5	80.4	19.8	18.3	0.9	20.8	18.4	37.1
PyCDA [52]		86.7	24.8	80.9	21.4	27.3	**30.2**	26.6	21.1	**86.6**	28.9	58.8	53.2	17.9	80.4	18.8	22.4	4.1	9.7	6.2	37.2
CrDoCo [62]		89.1	33.2	80.1	26.9	25.0	18.3	23.4	12.8	77.0	29.1	72.4	55.1	20.2	79.9	22.3	19.5	1.0	20.1	18.7	38.1
BDL [34]		89.2	40.9	81.2	29.1	19.2	14.2	29.0	19.6	83.7	35.9	80.7	54.7	23.3	82.7	**25.8**	28.0	2.3	**25.7**	19.9	41.3
FDA [39]		86.1	35.1	80.6	30.8	20.4	27.5	30.0	**26.0**	82.1	30.3	73.6	52.5	21.7	81.7	24.0	30.5	**29.9**	14.6	24.0	42.2
FADA [98]		**92.3**	**51.1**	**83.7**	33.1	**29.1**	28.5	28.0	21.0	82.6	32.6	**85.3**	55.2	**28.8**	**83.5**	24.4	**37.4**	0.0	21.1	15.2	43.8
Ours		90.1	46.7	82.7	**34.2**	25.3	21.3	**33.0**	22.0	84.4	**41.4**	78.9	**55.5**	25.8	83.1	24.9	31.4	20.6	25.2	**27.8**	**44.9**
AdaptSegNet [19]	ResNet101	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
CLAN [57]		87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
IntraDA [36]		90.6	37.1	82.6	30.1	19.1	29.5	32.4	20.6	**85.7**	40.5	79.7	58.7	31.1	86.3	31.5	48.3	0.0	30.2	35.8	46.3
MaxSquare [108]		89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	85.3	39.4	78.2	63.0	22.9	84.6	**36.4**	43.0	5.5	34.7	33.5	46.4
BDL [34]		91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	**43.6**	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
FADA [98]		**92.5**	47.5	**85.1**	**37.6**	**32.8**	33.4	33.8	18.4	85.3	37.7	**83.5**	**63.2**	**39.7**	**87.5**	32.9	47.8	1.6	**34.9**	39.5	49.2
FDA [39]		**92.5**	**53.3**	82.4	26.5	27.6	**36.4**	**40.6**	**38.9**	82.3	39.8	78.0	62.6	34.4	84.9	34.1	**53.1**	**16.9**	27.7	**46.4**	**50.4**
Ours		91.3	46.0	84.5	34.4	29.7	32.6	35.8	36.4	84.5	43.2	83.0	60.0	32.2	83.2	35.0	46.7	0.0	33.7	42.2	49.2

learning rate for DeepLab-ResNet101 and CD-SAM are 2.5e-4 and 1e-4, respectively, and are decreased by the same polynomial policy with power 0.9. For the discriminator, we use an Adam optimizer with momentum (0.9, 0.99). Its initial learning rate is set to 1e-6 for DeepLab-VGG16 and 1e-4 for DeepLab-ResNet101, respectively. We set λ to 0.0001 and 0.001 for DeepLab-VGG16 and DeepLab-ResNet101, respectively.

3.4.3 Performance Comparison

3.4.3.1 GTA5 to Cityscapes

Our method is first evaluated by using GTA5 as the source domain and Cityscapes as the target domain. The performance is assessed on 19 common classes between these two datasets by following the same evaluation criterion in previous studies [34, 62]. Our method is compared with existing state-of-the-art models by using VGG16 and

ResNet101 as the base architectures. As shown in Table 3.1, our method competes favorably against other models. Specifically, we surpass the mean intersection-over-union (mIoU) of feature alignment-based [17, 81, 57] and curriculum-based methods [51] by a large margin. This observation indicates that simply aligning feature space and label distribution cannot fully transfer domain knowledge in semantic segmentation. Compared to the models [18, 62, 34] that are based on image-to-image translation, our method gains up to 9.5% improvement by using VGG16, revealing that domain discrepancy can be further reduced by considering context adaptation. Similar to [19, 34], we also adapt structured output space in our model, but our method achieves significant performance improvement. This observation reveals the important role of context adaptation in knowledge transfer. It is noteworthy that the prediction of the "train" class is extremely challenging, owing to the limited "train" samples in the source domain. Our method enables to alleviate this limitation by adapting cross-domain context information. Compared to the CyCADA [18], we achieve 16.1% improvement on the "train" class.

3.4.3.2 SYNTHIA to Cityscapes

The superiority of our method is further proved on "SYNTHIA to Cityscapes". It is noteworthy that domain adaptation on "SYNTHIA to Cityscapes" is more challenging than "GTA5 to Cityscapes", owing to the large domain gap between these two domains. Following [34], we consider the 16 and 13 common classes for VGG16 and ResNet101-based models, respectively. As summarized in Table 3.2, we achieve a performance improvement of 1.8% and 1.0% over BDL [34] with VGG16 and ResNet101 base architectures. One of the most significant difference between these two domains is that SYNTHIA has much more 'person' instances than Cityscapes, which makes it hard to transfer common knowledge of the class 'person' by simply aligning

Table 3.2: The performance comparison by adapting from SYNTHIA to Cityscapes. Two base architectures (i.e., VGG16 and ResNet101) are used in our study. The comparison is performed on 16 common classes for VGG16 and 13 common classes for ResNet101.

SYNTHIA to Cityscapes																		
	Architecture	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	bus	motorbike	bicycle	**mIoU**
DCAN [60]	VGG16	79.9	30.4	70.8	1.6	**0.6**	22.3	6.7	23.0	76.9	73.9	41.9	16.7	61.7	11.5	10.3	38.6	35.4
PyCDA [52]		80.6	26.6	74.5	2.0	0.1	18.1	**13.7**	14.2	80.8	71.0	48.0	19.0	72.3	22.5	12.1	18.1	35.9
DADA [54]		71.1	29.8	71.4	3.7	0.3	**33.2**	6.4	15.6	81.2	78.9	52.7	13.1	75.9	25.5	10.0	20.5	36.8
GIO-Ada [53]		78.3	29.2	76.9	**11.4**	0.3	26.5	10.8	17.2	81.7	81.9	45.8	15.4	68.0	15.9	7.5	30.4	37.3
TGCF-DA [91]		**90.1**	**48.6**	80.7	2.2	0.2	27.2	3.2	14.3	**82.1**	78.4	54.4	16.4	**82.5**	12.3	1.7	21.8	38.5
BDL [34]		72.0	30.3	74.5	0.1	0.3	24.6	10.2	25.2	80.5	80.0	54.7	23.2	72.7	24.0	7.5	44.9	39.0
FADA [98]		80.4	35.9	**80.9**	2.5	0.3	30.4	7.9	22.3	81.8	**83.6**	48.9	16.8	77.7	**31.1**	13.5	17.9	39.5
FDA [39]		84.2	35.1	78.0	6.1	0.4	27.0	8.5	22.1	77.2	79.6	55.5	19.9	74.8	24.9	**14.3**	40.7	40.5
Ours		73.0	31.1	77.1	0.2	0.5	27.0	11.3	**27.4**	81.2	81.0	**59.0**	**25.6**	75.0	26.3	10.1	**47.4**	**40.8**
SIBAN [56]	ResNet101	82.5	24.0	79.4	✗	✗	✗	16.5	12.7	79.2	82.8	58.3	18.0	79.3	25.3	17.6	25.9	46.3
CLAN [57]		81.3	37.0	80.1	✗	✗	✗	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	47.8
MaxSquare [108]		82.9	40.7	80.3	✗	✗	✗	12.8	18.2	82.5	82.2	53.1	18.0	79.0	31.4	10.4	35.6	48.2
IntraDA [36]		84.3	37.7	79.5	✗	✗	✗	9.2	8.4	80.0	**84.1**	57.2	23.0	78.0	38.1	20.3	36.5	48.9
DADA [54]		**89.2**	44.8	81.4	✗	✗	✗	8.6	11.1	81.8	84.0	54.7	19.3	79.7	40.7	14.0	38.8	49.8
BDL [34]		86.0	**46.7**	80.3	✗	✗	✗	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	51.4
FDA [39]		79.3	35.0	73.2	✗	✗	✗	19.9	24.0	61.7	82.6	**61.4**	31.1	83.9	40.8	**38.4**	**51.1**	**52.5**
FADA [98]		84.5	40.1	**83.1**	✗	✗	✗	**20.1**	**27.2**	**84.8**	84.0	53.5	22.6	**85.4**	**43.7**	26.8	27.8	**52.5**
Ours		82.5	42.2	81.3	✗	✗	✗	18.3	15.9	80.6	83.5	**61.4**	**33.2**	72.9	39.3	26.6	43.9	52.4

marginal distribution or structured output space [34]. In contrast, by considering context information explicitly, we bring 7.3% improvement compared to BDL on this class with ResNet101-based model. This result demonstrates the benefit of explicitly adapting cross-domain context dependencies in semantic segmentation, especially for two domains with significant differences.

3.4.4 Ablation Study

3.4.4.1 GTA5 to Cityscapes

By incorporating CD-SAM and CD-CAM individually, we get 2.4% and 2.3% performance boost over the VGG16-based baseline (Table 3.3). Taken them together, the mIoU is further improved to 44.9 mIoU. Similarly, 0.5% and 0.3% improvement is also observed in the ResNet101-based model by considering CD-SAM and CD-CAM. We achieve 49.2 mIoU by integrating both attention modules. To qualitatively demon-

Table 3.3: Ablation study on "GTA5 to Cityscapes".

GTA5 to Cityscapes			
Base	CD-SAM	CD-CAM	**mIoU**
VGG16			41.3
	✓		43.7
		✓	43.6
	✓	✓	44.9
ResNet101			48.5
	✓		49.0
		✓	48.8
	✓	✓	49.2

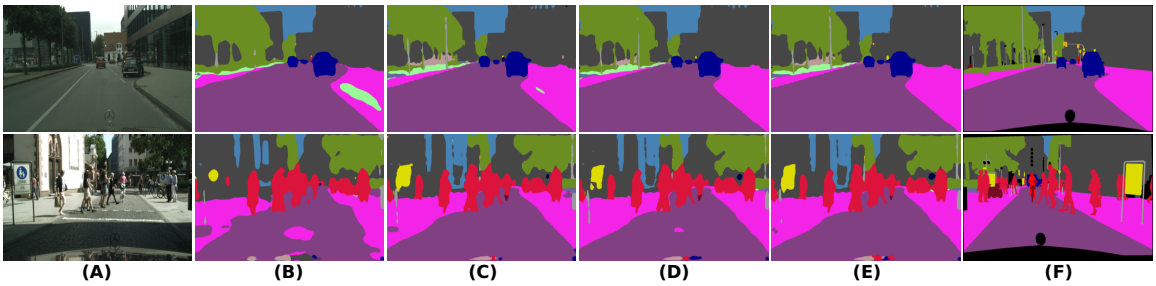

Figure 3.5: Qualitative comparison between our method and the baseline model BDL [34]. For each given image (A), we present its segmentation output from (B) BDL, (C) our method incorporating CD-SAM only, (D) our method incorporating CD-CAM only, (E) our method considering both CD-SAM and CD-CAM, and the ground truth (F).

strate the superiority of our method, we showcase the examples of its segmentation outputs at different stages in Figure 3.5. As shown in the figure, our method enables to predict more consistent segmentation outputs than the baseline model and becomes increasingly accurate by incorporating two cross-domain attention modules.

3.4.4.2 SYNTHIA to Cityscapes

For VGG16-based model, CD-SAM and CD-CAM contribute to 1.2% and 1.0% improvement compared to the baseline (Table 3.4). Our method gains 1.8% improvement by combining them. By applying CD-SAM and CD-CAM to ResNet101,

Table 3.4: Ablation study on "SYNTHIA to Cityscapes".

SYNTHIA to Cityscapes			
Base	CD-SAM	CD-CAM	**mIoU**
VGG16			39.0
	✓		40.2
		✓	40.0
	✓	✓	40.8
ResNet101			51.4
	✓		51.8
		✓	52.0
	✓	✓	52.4

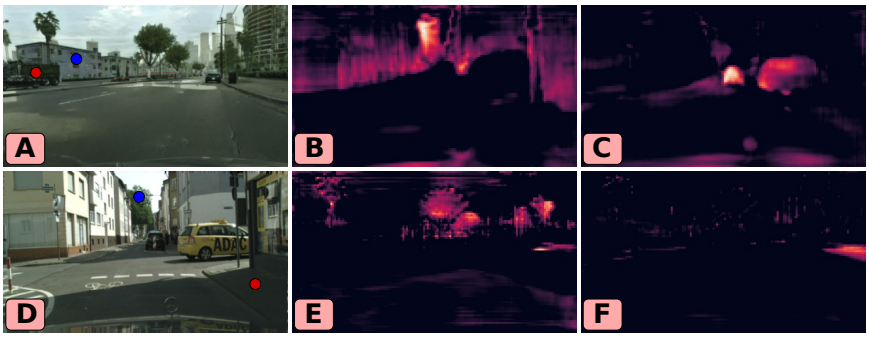

Figure 3.6: An example of the spatial attention map. Given a source image (A) and a target image (D), we present the source-to-target attention maps (B) and (C) for the blue and red point in (A), respectively. Similarly, we present the target-to-source attention maps (E) and (F) of the blue and red point in (D), respectively.

we achieve 51.8 and 52.0 mIoU with 0.4% and 0.6% improvement over the baseline, respectively. It is further boosted to 52.4 mIoU when both of them are considered. Our results reveal that the proposed cross-attention mechanism significantly contributes to domain adaptation in semantic segmentation by adapting context dependencies. Furthermore, the two cross-domain attention modules play a complementary role in capturing context information.

Table 3.5: Ablation study of λ_s, λ_t, ξ_s, and ξ_t.

$\lambda_s/\lambda_t/\xi_s/\xi_t$	0.1	1	10
mIoU	43.7	44.9	40.6

3.4.4.3 Visualization of the Cross Attention

To fully understand the cross-attention mechanism in our model, we visualize the spatial attention maps in this section. As shown in Figure 3.6, two images are randomly selected from the source and target domain. Recall that each position in the source feature has a spatial attention map corresponding to all positions in the target feature, and vice versa. We, therefore, select two positions in the source image and visualize their "source-to-target" attention map. For the blue point that is marked on a building in the source image (Figure 3.6 A), its spatial attention map (Figure 3.6 B) mainly corresponds to the building in the target image (Figure 3.6 D). For the red point that is marked on a truck in Figure 3.6 A, its spatial attention map (Figure 3.6 C) highlights the cars in Figure 3.6 D. Similarly, we select another two positions in the target image and conduct the visualization of the "target-to-source" attention map. For the blue point in the target image (Figure 3.6 D), its attention map (Figure 3.6 E) focuses on the vegetation in the source image (Figure 3.6 A). These visualizations demonstrate the power of our method in capturing cross-domain spatial context information.

3.4.4.4 Parameter Sensitivity Analysis

In this section, we perform a sensitivity analysis of λ_s, λ_t, ξ_s, and ξ_t as shown in Table 3.5. We investigate three different choices, *i.e.*, 0.1, 1, and 10, indicating how much attention should pay for the context information from the opposite domain. Our results reveal that $\lambda_s = \lambda_t = \xi_s = \xi_t = 1$ performs best. The reason is that a

small value fails to capture cross-domain context dependencies, while a large value may disturb the original feature. In addition, by setting $\lambda_s = \lambda_t = 0.1$, $\xi_s = \xi_t = 1$, we have mIoU 43.2. We also evaluate the scenario where λ_s, λ_t, ξ_s, and ξ_t are learnable hyperparameters, which gives rise to mIoU 44.0.

3.5 Summary and Discussion

In this chapter, we propose an innovative cross-attention mechanism for domain adaptation by adapting the semantic context. Specifically, we introduce two cross-domain attention modules to capture spatial and channel context between source and target domains. The obtained contextual dependencies, which are shared across two domains, are further adapted to decrease the domain discrepancy. Empirical studies demonstrate that our method achieves the new state-of-the-art performance on "GTA5-to-Cityscapes" and "SYNTHIA-to-Cityscapes".

Despite the impressive performance achieved by various UDA methods, recent studies imply that deep neural networks are vulnerable to adversarial attacks, i.e., inputs with a slight but intentional perturbation are incorrectly classified by the network. Such vulnerability makes it risky for some security-related applications (e.g., semantic segmentation in autonomous cars) and triggers tremendous concerns about the model reliability. Unfortunately, the robustness of existing UDA methods remains unexplored. We address this limitation in the next chapter.

CHAPTER 4

Exploring Robustness of Unsupervised Domain Adaptation in Semantic Segmentation

In this chapter, we comprehensively evaluate the robustness of existing UDA methods and propose a robust UDA approach. It is rooted in two observations: i) the robustness of UDA methods in semantic segmentation remains unexplored, which poses a security concern in this field; and ii) although commonly used self-supervision (e.g., rotation and jigsaw) benefits model robustness in classification and recognition tasks, they fail to provide the critical supervision signals that are essential in semantic segmentation. These observations motivate us to propose adversarial self-supervision UDA (or ASSUDA) that maximizes the agreement between clean images and their adversarial examples by a contrastive loss in the output space. Extensive empirical studies on commonly used benchmarks demonstrate that ASSUDA is resistant to adversarial attacks.

4.1 Introduction

Semantic segmentation aims to predict semantic labels of each pixel in the given images, which plays an important role in autonomous driving [109] and medical diagnosis [110]. However, pixel-wise labeling is extremely time-consuming and labor-intensive. For instance, 90 minutes are required to annotate a single image for the Cityscapes dataset [66]. Although synthetic datasets [49, 50] with freely available labels provide an opportunity for model training, the model trained on synthetic data suffers from dramatic performance degradation when applying it directly to the real data of interest.

Motivated by the success of unsupervised domain adaptation (UDA) in image classification, various UDA methods for semantic segmentation are recently proposed. The key idea of these methods is to learn domain-invariant representations by minimizing marginal distribution distance between the source and target domains [17], adapting structured output space [19, 89], or reducing appearance discrepancy through image-to-image translation [18, 59, 34]. Another alternative is to explicitly explore the supervision signals from the target domain through self-training. The key idea is to alternatively generate pseudo labels on target data and re-train the model with these labels. Most of the existing state-of-the-art UDA methods in semantic segmentation rely on this strategy and demonstrate significant performance improvement. [55, 34, 39, 46, 111].

However, one of the critical issues of the aforementioned UDA methods is that they are possibly vulnerable to adversarial attacks. In other words, the performance of a UDA model may dramatically degrade under an unnoticeable perturbation. Unfortunately, the robustness of UDA methods remains largely unexplored in the literature. With the increasing applications of UDA methods in security-related areas, the lack of robustness of these methods leads to massive safety concerns. For instance, even small-magnitude perturbations on traffic signs can potentially cause disastrous consequences to autonomous cars [112, 113], such as life-threatening accidents.

Self-supervised learning (SSL) aims to learn more transferable and generalizable features for vision tasks (*e.g.*, classification and recognition) [114, 115, 116, 117]. Key to SSL is the design of pretext tasks, such as rotation prediction, selfie, and jigsaw, to obtain self-derived supervisory signals on unlabeled data. Recent studies reveal that SSL is effective in improving model robustness and uncertainty [118]. However, commonly used pretext tasks are designed to capture the global representation of a given image or an image patch. Such pretext tasks fail to provide critical supervision

signals for segmentation tasks where fine-grained or pixel-level representations are required [119].

In this chapter, we first perform a comprehensive study to evaluate the robustness of existing UDA methods in semantic segmentation. Our results reveal that these methods can be easily fooled by small perturbations and show dramatic performance degradation. To remedy this problem, we introduce a new UDA method known as ASSUDA to robustly adapt domain knowledge in urban-scene semantic segmentation. The key insight of our method is to leverage the regularization power of adversarial examples. Specifically, we propose the adversarial self-supervision that maximizes the agreement between clean images and their adversarial examples by a contrastive loss in the output space. The adversarial examples aim to i) provide fine-grained supervision signals for unlabeled target data, so that more transferable and generalizable features can be learned and ii) improve the robustness of our model against adversarial attacks by taking advantage of both adversarial training and self-supervision.

Our main contributions can be summarized as i) To the best of our knowledge, this chapter presents the first systematic study on how existing UDA methods in semantic segmentation are vulnerable to adversarial attacks. We believe this investigation provides new insight into this area; ii) We propose a new UDA method that takes advantage of adversarial training and self-supervision to improve the model robustness; iii) Comprehensive empirical studies demonstrate the robustness of our method against adversarial attacks on two benchmark settings, *i.e.*, "GTA5 to Cityscapes" and "SYNTHIA to Cityscapes".

4.2 Related Work

4.2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) refers to the scenario where no labels are available for the target domain. In the past few years, various UDA methods are proposed for semantic segmentation, which can be mainly summarized as three streams: i) adapt domain-invariant features by directly minimizing the representation distance between two domains [17, 32]; ii) align pixel space through translating images from the source domain to the target domain [18, 33]; iii) align structured output space, which is inspired by the fact that source output and target output share substantial similarities in terms of structure layout [19]. However, simply aligning cross-domain distribution has limited capability in transferring pixel-level domain knowledge for semantic segmentation. To address this problem, the most recent studies integrate self-training into existing UDA frameworks and demonstrate the state-of-the-art performance [55, 34, 39, 46].

Our method instead resorts to self-supervision by integrating contrastive learning into existing UDA methods. This strategy demonstrates two advantages: i) provides supervision for the target domain, which is proved to be robust to the label corruption; ii) encourages the model to learn more transferable and robust features. Another major difference is that our method mainly focuses on improving model robustness against adversarial attacks, which is overlooked by existing UDA methods.

4.2.2 Self-supervised Learning

Self-supervision aims to make use of massive amounts of unlabeled data through getting free supervision from the data itself. This is typically achieved by training self-supervised tasks (a.k.a., pretext tasks) through two paradigms, *i.e.*, pre-training & fine-tuning and multi-task learning. Specifically, the pre-training & fine-tuning

first performs pre-training on the pretext task, then fine-tunes on the downstream task. In contrast, multi-task learning optimizes the pretext task and the downstream task simultaneously. Our method falls into the latter, where the downstream task is to predict the segmentation labels of the target domain. To learn transferable and generalizable features through self-supervision, it is essential to design pretext tasks that are tailored to the downstream task. Commonly used pretext tasks include exemplar [114], rotation [115], predicting the relative position between two random patches [120], and jigsaw [121]. Motivated by this, recent UDA methods introduce self-supervision into segmentation adaptation to learn domain invariant feature representations [122, 123]. Although these commonly used pretext tasks contribute to cross-domain feature alignment, they are mainly designed to capture the global feature, and therefore have limited capability in learning fine-grained representations that are essential in semantic segmentation.

By contrast, this chapter proposes to use adversarial examples to build pretext tasks. Specifically, we maximize agreement between each image and its adversarial example via a contrastive loss in the output space. This is different from [117] that performs contrastive learning in the latent space. Furthermore, rather than focus on single-domain tasks [124, 125], our method is tailored to UDA environments to adapt domain knowledge and improve robustness simultaneously. Therefore, i) our method is encouraged to learn more transferable features which are domain-invariant and fine-grained; ii) the trained model is more robust to label corruption and adversarial attacks. Another closely related work is [126] which shares a similar spirit with us but with clear differences: i) rather than perturb the intermediate feature maps, we perform the perturbation to the input images; ii) we target on improving model robustness, instead of the segmentation accuracy on clean images.

4.2.3 Adversarial Attacks

Previous studies reveal that adversarial attacks are commonly observed in machine learning methods such as SVMs [127] and logistic regression [128]. Recent publications suggest that neural networks are also highly vulnerable to adversarial perturbations [129, 130]. Even worse, adversarial attacks are proven to be transferable across different models [131], *i.e.*, the adversarial examples generated to attack a specific model are also harmful to other models. To fully understand adversarial attacks in deep neural networks (DNNs), considerable attention is received in the past few years. Specifically, [130] proposes a fast gradient sign method (FGSM) to efficiently generate adversarial examples with only one gradient step. DeepFool [132] generates minimal perturbations by iteratively linearizing the image classifier. By utilizing the differential evolution, [133] enables us to generate one-pixel adversarial perturbations to accurately attack DNNs.

Unlike the aforementioned studies that focus on effectively creating adversarial attacks, our method uses adversarial examples to build pretext tasks for UDA models, and in turn to improve the model robustness. This is motivated by the fact that a clean image and its adversarial example should have the same segmentation output. Therefore, we can get supervision for free and encourage our method to learn discriminative representation for segmentation tasks.

4.3 Methodology

We first briefly recall the preliminary of UDA, adversarial training, and self-supervision. We then perform the first-of-its-kind empirical study to show that existing UDA methods are vulnerable to adversarial attacks, which arises tremendous concerns for the application of these methods in safety-critical areas. To address this problem,

we propose a new domain adaptation method known as ASSUDA to improve the model robustness without satisfying much predictive accuracy. Specifically, our method takes advantage of adversarial training and self-supervision and thus enabling us to generate more robust and generalizable features.

4.3.1 Preliminary

4.3.1.1 UDA in Semantic Segmentation

Consider the problem of UDA in semantic segmentation, where a labeled source domain $\mathcal{X}_s\{(x_s^{(i)}, y_s^{(i)})\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{X}_t\{x_t^{(j)}\}_{j=1}^{n_t}$ are given. Our goal is to learn a segmentation model $f_{\theta_C}(\cdot)$ which guarantees accurate prediction on the target domain. Formally, the loss function of a typical UDA model is defined as:

$$\mathcal{L}_{seg}(x_s, y_s; \theta_C) + \alpha \mathcal{L}_{dis}(x_s, x_t), \tag{4.1}$$

where $\mathcal{L}_{seg}$ is the typical segmentation objective, $\mathcal{L}_{dis}$ measures the domain distance. The most commonly used $\mathcal{L}_{dis}$ is the adversarial loss $\mathcal{L}_{adv}$ that encourages a discriminative and domain-invariant feature representation through a domain discriminator $D_{\theta_D}(\cdot)$ [17, 18, 19], which is formalized as:

$$\begin{aligned} \mathcal{L}_{adv}(x_s, x_t; \theta_C, \theta_D) = \ & \mathbb{E}[log D_{\theta_D}(f_{\theta_C}(x_s))] + \\ & \mathbb{E}[log(1 - D_{\theta_D}(f_{\theta_C}(x_t)))] \end{aligned} \tag{4.2}$$

4.3.1.2 Adversarial Training

Recall that the objective of the vanilla adversarial training is:

$$\arg\min_x \mathbb{E}_{(x,y)\sim\mathbb{D}}[\max_{\eta\in\mathbb{S}} \mathcal{L}(f_\theta(x+\eta), y)] \tag{4.3}$$

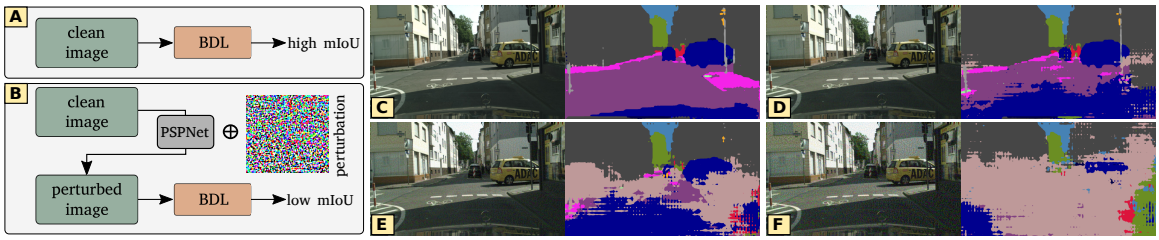

Figure 4.1: Robustness study of BDL [34] on "GTA5 to Cityscapes" with backbone ResNet101. (A) the traditional paradigm uses clean test data to evaluate the performance of BDL; (B) we use PSPNet as the surrogate model to generate perturbed test data which are then used to evaluate BDL; (C) a clean image and its segmentation output predicted by BDL; (D), (E), and (F) indicate the perturbed images of (C) with $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$, respectively, along with their BDL predictions. Although the perturbations are unnoticeable, they can easily deceive BDL, resulting in dramatic performance degradation.

Base	ϵ	GTA5 to City	SYNTHIA to City
VGG16	0.1	41.3 $\rightarrow$ 30.5	39.0 $\rightarrow$ 29.3
	0.25	41.3 $\rightarrow$ 14.6	39.0 $\rightarrow$ 13.6
	0.5	41.3 $\rightarrow$ 7.10	39.0 $\rightarrow$ 5.90
ResNet101	0.1	48.5 $\rightarrow$ 36.2	51.4 $\rightarrow$ 41.2
	0.25	48.5 $\rightarrow$ 19.9	51.4 $\rightarrow$ 26.6
	0.5	48.5 $\rightarrow$ 6.50	51.4 $\rightarrow$ 11.0

Table 4.1: Performance of pre-trained BDL on clean test data vs perturbed test data. Three sets of perturbed data are generated with $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$, respectively.

where $\mathbb{S}$ are allowed perturbations, $\tilde{x} \leftarrow x + \eta$ is an adversarial example of x with the perturbation η. To obtain η, the most commonly used attack method is FGSM [130]:

$$\eta = \epsilon \, \text{sign}(\bigtriangledown_x \mathcal{L}(f_\theta(x), y)), \tag{4.4}$$

where ϵ is the magnitude of the perturbation. The generated adversarial examples $\tilde{x}$ are imperceptible to human but can easily fool deep neural networks. Recent studies further prove that training models exclusively on adversarial examples can improve the model robustness [134].

4.3.2 Robustness of UDA Methods

Although existing UDA methods achieve record-breaking predictive accuracy, their robustness against adversarial attacks remains unexplored. We hypothesis that they are also vulnerable to adversarial attacks, which makes it risky to apply them in safety-critical scenarios. To fill this gap and to validate our hypothesis, we perform black-box attacks on BDL [34] by conducting the following two steps: 1) for each clean image in the test data, we first generate its adversarial example by attacking PSPNet [69] with $\epsilon = 0.1$, $\epsilon = 0.25$ and $\epsilon = 0.5$, respectively; 2) we then evaluate the pre-trained BDL model on the generated adversarial examples (or perturbed test data) (Figure 4.1). The rationale behind this setting is that i) recent state-of-the-art UDA methods in semantic segmentation [37, 38, 39, 46, 111] share similar spirits with BDL, so conducting pilot studies on this method would be representative; ii) a black-box attack assumes that the attacker can only access very limited information of the victim model, which is a common case in the real world. Therefore, a black-box attack would be very dangerous if it can work; iii) adversarial attacks are transferable across different models [130], *i.e.*, the adversarial examples generated to attack a surrogate model are also harmful to other models. We hereby perform the black-box attack to examine the transferability of adversarial examples on UDA models.

As shown in Table 4.1, despite the remarkable performance of BDL on the clean test data, even slight and unnoticeable perturbations can result in dramatic performance degradation. For instance, BDL (with VGG16 backbone) only achieves a mean IoU (mIoU) of 30.5% on the perturbed test data generated by $\epsilon = 0.1$, compared to 41.3% on the clean data. By increasing the perturbation ratio ϵ, the performance can drop even further (Figure 4.1), indicating that BDL can be easily fooled by slight perturbations on the test data, even though the perturbation is generated by a surrogate model. This empirical study suggests that existing UDA methods are

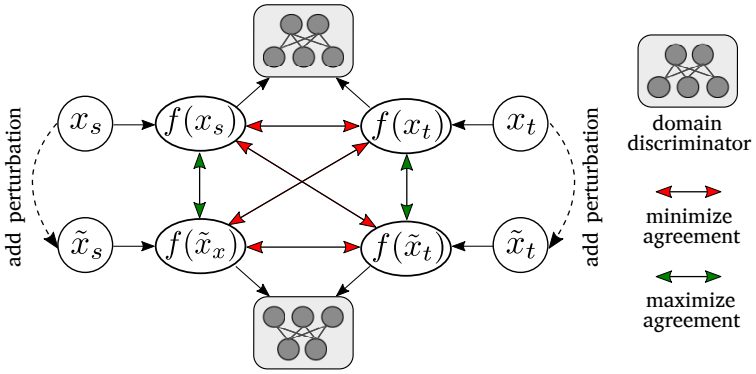

Figure 4.2: An overview of the proposed method. For each sampled pair of source image x_s and target image x_t, we generate their adversarial example $\tilde{x}_s$ and $\tilde{x}_t$, respectively. A segmentation model $f(\cdot)$ and a domain discriminator are trained to maximize/minimize agreement and align cross-domain representations.

also possibly vulnerable to adversarial perturbations, which can make them especially risky for some security-related areas.

4.3.3 Adversarial Self-Supervision UDA

To address this problem, the most straightforward approach is adversarial training (equation 4.3) which requires class labels to generate adversarial examples. However, we are unable to access the labels of target data under the scenario of UDA (equation 4.1). The success of existing UDA methods heavily relies on the self-training strategy that alternatively generates highly confident pseudo labels for the target domain and re-trains the model using these labels [34, 38, 46, 111, 20]. Although pseudo labels provide an opportunity to generate adversarial examples for the target data, these labels are usually noisy and less accurate. Hendrycks *et al.*prove that self-supervision improves the robustness of deep neural networks for vision tasks [118]. Nevertheless, commonly used pretext tasks (*e.g.*, rotation prediction and jigsaw) model

global representation and fail to provide the critical supervision signals in learning discriminative features for semantic segmentation.

Algorithm 1: The whole training process.

Input: Source data $\{\mathcal{X}_s, Y_s\}$ and target data $\{\mathcal{X}_t\}$,
segmentation model initialized as θ_C,
domain discriminator initialized as θ_D,
batch size N, number of training iteration R

Result: θ_C and θ_D

for $r \leftarrow 1$ **to** R **do**

Sample a batch of source-target pairs $\{x_s^{(k)}, x_t^{(k)}\}_{k=1}^{N}$

 # adversarial attack

for $k \in \{1, ..., N\}$ **do**

Generate adversarial examples: $\{\tilde{x}_s^{(k)}, \tilde{x}_t^{(k)}\}_{k=1}^{N}$

Define $x^{(4k-3)} = x_s^{(k)}, x^{(4k-2)} = x_t^{(k)}, x^{(4k-1)} = \tilde{x}_s^{(k)}, x^{(4k)} = \tilde{x}_t^{(k)}$

end

 # adversarial self-supervision

for $i \in \{1, ..., 4N\}$ *and* $j \in \{1, ..., 4N\}$ **do**

$s_{i,j} = exp(\frac{-dist\left(f_{\theta_C}(x^{(i)}), f_{\theta_C}(x^{(j)})\right)}{2\sigma^2})$

end

Define $\ell_{i,j} = -log\frac{exp(s_{i,j})}{\sum_{k=1}^{4N} \mathbb{1}_{[k \neq i]} exp(s_{i,k})}$

 # contrastive loss

$\mathcal{L}_{con} = \frac{1}{4N} \sum_{k=1}^{N} [\ell_{4k-3,4k-1} + \ell_{4k-1,4k-3} + \ell_{4k-2,4k} + \ell_{4k,4k-2}]$

 # update model parameters

$\theta_C \leftarrow \theta_C - \beta \bigtriangledown_{\theta_C} \mathcal{L}_{total}$

$\theta_D \leftarrow \theta_D - \lambda \bigtriangledown_{\theta_D} \mathcal{L}_{total}$

end

return θ_C and θ_D

These challenges raise the question: *can we take advantage of both adversarial training and self-supervision in improving the robustness of UDA methods in semantic segmentation?* To answer this question, we propose to build a pretext task by using adversarial examples (Figure 4.2). Specifically, we consider a clean image and its

adversarial example as a positive pair and maximize agreement on their segmentation outputs by a contrastive loss. This is motivated by the fact that a clean image and its adversarial example should share the same segmentation map. Different from [117] that uses a contrastive loss in the latent space, our pretext task is performed in the output space to learn discriminative representations for semantic segmentation. To adapt knowledge from the source domain to the target domain, a domain discriminator is applied to the source and target outputs. It is worth mentioning that the domain discriminator minimizes the domain-level difference, while the contrastive loss is performed on the pixel level.

Our model is built upon BDL [34] that generates the transformed source images $\mathcal{X}_{s\to t}$ and pseudo labels $Y_{t'}$ of $\mathcal{X}_t$. For simplicity, we use $\mathcal{X}_s$ to represent $\mathcal{X}_{s\to t}$ in the remaining of this chapter, unless otherwise specified. At each training iteration r, a minibatch of N source-target pairs are randomly sampled from $\mathcal{X}_s$ and $\mathcal{X}_t$, resulting in $2N$ examples: $\{x_s^{(i)}, x_t^{(i)}\}_{i=1}^N$. Their adversarial examples $\{\tilde{x}_s^{(i)}, \tilde{x}_t^{(i)}\}_{i=1}^N$ are generated by:

$$\begin{aligned}\tilde{x}_s^{(i)} &= x_s^{(i)} + \epsilon_m \text{sign}(\bigtriangledown_x[\mathcal{L}_{seg}(x_s^{(i)}, y_s^{(i)}; \theta_C)]) \\ \tilde{x}_t^{(i)} &= x_t^{(i)} + \epsilon_m \text{sign}(\bigtriangledown_x[\mathcal{L}_{seg}(x_t^{(i)}, y_{t'}^{(i)}; \theta_C)])\end{aligned} \tag{4.5}$$

where ϵ_m is the training perturbation magnitude.

Given these $4N$ data points $\{x_s^{(i)}, x_t^{(i)}, \tilde{x}_s^{(i)}, \tilde{x}_t^{(i)}\}_{i=1}^N$, each pair of examples $\{x_\alpha^{(i)}, \tilde{x}_\alpha^{(i)}\}$ is considered as a positive pair (α can be either s or t to denote a source or a target domain), while the other $4N-2$ examples are considered as negative examples. We define the contrastive loss for a positive pair (i, j) as

$$\ell_{i,j} = -log\frac{exp(sim(f_{\theta_C}(x^{(i)}), f_{\theta_C}(x^{(j)})))}{\sum_{k=1}^{4N} \mathbb{1}_{[k\neq i]} exp(sim(f_{\theta_C}(x^{(i)}), f_{\theta_C}(x^{(k)})))}, \tag{4.6}$$

where $sim(\mathbf{U}, \mathbf{V}) = exp(-dist(\mathbf{U}, \mathbf{V})/(2\sigma^2))$ is Gaussian kernel that is used to measure the similarity between two segmentation output tensors $\mathbf{U}$ and $\mathbf{V}$, $dist(;)$ is the

Euclidean distance. The contrastive loss $\mathcal{L}_{con}(x_s, \tilde{x}_s, x_t, \tilde{x}_t; \theta_C))$ is computed across all positive pairs (see Algorithm 1). Taken together, the training objective of our goal is $\min_{\theta_C} \max_{\theta_D} \mathcal{L}_{total}$, where $\mathcal{L}_{total}$ is:

$$\begin{aligned}\mathcal{L}_{total} = &\mathcal{L}_{seg}(x_s, y_s; \theta_C) + \mathcal{L}_{seg}(\tilde{x}_s, y_s; \theta_C) + \\ &\mathcal{L}_{seg}(x_t, y_{t'}; \theta_C) + \mathcal{L}_{seg}(\tilde{x}_t, y_{t'}; \theta_C) + \\ &\gamma \mathcal{L}_{adv}(x_s, x_t; \theta_C, \theta_D) + \\ &\gamma \mathcal{L}_{adv}(\tilde{x}_s, \tilde{x}_t; \theta_C, \theta_D) + \\ &\delta \mathcal{L}_{con}(x_s, \tilde{x}_s, x_t, \tilde{x}_t; \theta_C),\end{aligned} \tag{4.7}$$

where δ and γ are two hyper-parameters. Therefore, our model can leverage the regularization power of adversarial examples through a self-supervision manner, and in turn, improve the model robustness against adversarial attacks. The whole training process is detailed in Algorithm 1.

4.4 Experiments

4.4.1 Datasets

Following the same setting as previous studies, we use GTA5 [49] and SYNTHIA-RAND-CITYSCAPES [50] as the source domain, and use Cityscapes [66] as the target domain. GTA5 is composed of 24,966 images (resolution: 1914 × 1052) with pixel-level semantic labels, which are collected from a photo-realistic open-world game known as Grand Theft Auto V. SYNTHIA-RAND-CITYSCAPES dataset is generated from a virtual city, including 9,400 images (resolution: 1280 × 760) with precise pixel-level semantic annotations. Cityscapes is a large-scale street scene dataset collected from 50 cities. A total of 5,000 images (resolution: 2048 × 1024) are contained in Cityscapes, with 2,975 training images, 500 validation images, and 1,525 test images. We follow

GTA5 to Cityscapes																							
	ϵ	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bicycle	**mIoU**	**mIoU drop**	**mIoU***
FDA [39]		73.9	18.5	69.7	7.5	6.4	**18.7**	23.9	21.5	76.7	12.2	66.3	45.2	18.4	70.2	18.9	13.9	**14.6**	9.3	22.0	32.0	10.2	42.4
AdaptSegNet [19]		71.9	22.7	70.8	7.6	7.9	16.5	15.4	8.3	71.8	12.2	52.6	33.8	0.6	65.8	15.8	7.6	0.0	0.7	0.1	25.4	9.6	35.0
PCEDA [135]	0.1	**90.9**	25.0	73.5	6.3	7.2	14.2	**24.0**	**27.4**	76.2	23.4	70.3	45.0	19.9	70.0	16.3	20.3	0.0	9.8	**25.1**	33.4	11.2	44.6
BDL [34]		64.0	21.9	70.0	10.0	3.9	8.4	20.5	12.8	77.4	22.3	79.2	49.8	13.8	73.2	17.8	12.1	0.0	7.8	15.2	30.5	10.8	41.3
Ours		90.6	**41.5**	**80.1**	**22.6**	**10.4**	15.4	23.0	16.0	**82.7**	**34.9**	**81.6**	**52.5**	**23.9**	**82.2**	**22.5**	**21.9**	7.0	**15.4**	21.4	**39.3**	**0.4**	39.7
FDA		25.4	3.4	24.5	0.5	1.6	2.4	7.7	6.4	58.6	1.2	44.8	6.5	1.4	14.6	4.9	0.4	0.1	0.1	1.3	10.8	31.4	42.4
AdaptSegNet		5.4	5.0	43.8	1.2	2.2	3.7	6.3	2.5	31.3	3.9	22.8	6.2	0.0	11.9	4.3	0.1	0.0	0.0	0.0	7.9	27.1	35.0
PCEDA	0.25	34.6	1.5	40.9	0.6	1.6	2.2	9.6	11.1	56.4	0.5	43.8	12.7	2.0	28.0	7.0	3.7	0.0	1.0	5.0	13.8	30.8	44.6
BDL		25.4	4.7	55.1	2.8	1.5	1.3	9.1	4.3	61.3	1.5	54.1	26.7	0.1	20.7	6.5	1.5	0.0	0.7	1.0	14.6	26.7	41.3
Ours		**89.7**	**30.4**	**78.2**	**13.4**	**11.4**	**11.1**	**19.4**	**14.5**	**79.2**	**27.0**	**84.8**	**49.7**	**19.0**	**78.6**	**17.1**	**18.1**	**3.0**	**7.2**	**17.2**	**35.2**	**4.5**	39.7
FDA		22.0	0.4	3.2	0.0	1.3	0.1	1.9	0.6	33.8	1.1	22.6	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	4.6	37.6	42.4
AdaptSegNet		0.1	0.0	14.4	0.0	2.1	0.7	2.9	0.4	23.3	0.0	8.4	0.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	2.8	32.2	35.0
PCEDA	0.5	26.8	0.1	15.0	0.1	1.3	0.1	2.5	2.3	18.1	0.0	15.4	0.1	0.0	2.0	0.2	0.0	0.0	0.0	0.0	4.4	40.2	44.6
BDL		27.8	0.9	36.8	0.5	1.2	0.1	2.7	0.9	34.1	0.0	25.1	5.4	0.0	0.7	0.0	0.0	0.0	0.0	0.0	7.1	34.2	41.3
Ours		**75.7**	**11.7**	**66.1**	**2.7**	**6.0**	**3.7**	**13.6**	**8.6**	**66.8**	**14.0**	**79.1**	**37.2**	**4.0**	**59.0**	**7.2**	**9.6**	**0.4**	**0.1**	**6.0**	**24.8**	**14.9**	39.7
FDA [39]		85.8	27.8	70.2	8.6	7.4	17.9	30.7	23.4	70.8	22.4	59.7	53.8	26.5	71.6	29.2	26.8	**6.3**	23.1	**38.3**	36.9	13.5	50.4
FADA [98]		53.2	19.7	65.2	6.3	14.1	21.3	19.0	8.2	74.4	21.6	55.7	50.3	14.8	73.2	13.4	9.1	1.0	9.6	20.5	29.0	20.2	49.2
IntraDA [36]		89.1	31.1	76.6	11.3	16.4	14.9	25.3	15.8	80.8	29.4	74.9	54.3	23.3	78.7	32.1	**39.2**	0.0	21.5	30.8	39.2	7.1	46.3
CLAN [57]		75.8	21.3	69.8	11.9	7.3	12.7	24.6	8.8	77.1	20.4	66.9	51.0	19.6	65.4	28.7	31.3	2.5	15.2	24.8	33.4	9.8	43.2
MaxSquare [108]	0.1	28.6	9.3	52.0	3.9	3.1	9.7	29.1	10.3	73.6	10.2	41.7	46.1	19.1	36.1	26.5	10.7	0.2	17.2	28.0	24.0	22.4	46.4
AdaptSegNet [19]		80.9	21.2	66.3	7.4	5.7	7.4	25.2	6.5	76.2	12.5	69.9	45.6	11.7	71.3	21.8	8.0	1.6	6.5	14.3	29.5	12.9	42.4
PCEDA [135]		**89.8**	31.8	75.8	17.4	9.2	26.9	**31.1**	30.0	80.0	19.3	**85.6**	**55.2**	**27.5**	79.4	30.2	34.4	0.0	20.3	**38.3**	41.2	9.3	50.5
BDL [34]		75.5	31.3	75.3	8.8	8.5	17.1	29.3	23.0	76.9	22.4	80.5	51.2	25.8	51.9	24.0	33.3	1.6	20.3	31.3	36.2	12.3	48.5
Ours		89.3	**37.7**	**81.3**	**21.0**	**18.3**	**28.6**	29.0	**31.4**	**81.8**	**33.9**	82.2	51.9	25.9	**80.4**	**34.9**	31.3	0.0	**30.4**	33.1	**43.3**	**0.6**	43.9
FDA		50.8	6.7	51.0	1.6	3.7	3.5	17.2	6.3	49.5	1.5	60.9	28.3	12.8	49.1	14.5	4.6	1.2	2.6	25.0	20.6	29.8	50.4
FADA		54.1	14.8	50.4	2.2	8.2	6.8	4.7	0.9	59.4	7.4	32.8	29.9	3.0	53.6	4.1	0.3	1.2	0.7	5.9	17.9	31.3	49.2
IntraDA		26.4	3.0	46.3	0.4	4.5	0.7	8.6	0.5	30.9	0.4	43.9	21.3	1.2	47.5	8.33	7.5	0.0	0.2	6.5	13.6	32.7	46.3
CLAN		58.3	9.4	52.7	5.0	2.7	1.3	14.7	2.1	58.5	3.0	64.5	37.6	14.0	46.1	20.0	13.6	**1.8**	3.6	17.3	22.4	20.8	43.2
MaxSquare	0.25	15.2	2.3	37.9	2.7	1.5	1.0	15.8	1.8	54.1	1.5	30.6	14.3	7.2	31.5	11.8	1.6	0.0	0.7	13.8	12.9	33.5	46.4
AdaptSegNet		66.9	4.8	32.8	1.3	2.4	0.7	13.2	1.2	60.6	2.4	65.3	19.6	1.5	49.0	8.2	1.2	0.0	0.1	0.8	17.5	24.9	42.4
PCEDA		76.4	3.0	50.9	1.5	3.3	11.5	18.1	10.0	59.3	0.6	59.4	37.0	16.1	49.6	11.6	5.6	0.0	2.6	25.2	23.3	27.2	50.5
BDL		40.7	7.2	56.6	3.1	2.0	4.0	20.3	5.5	62.7	1.5	65.8	19.4	15.3	30.2	8.0	8.4	0.0	6.4	21.2	19.9	28.6	48.5
Ours		**87.9**	**26.6**	**75.0**	**11.1**	**12.5**	**24.4**	**26.0**	**28.3**	**74.2**	**19.5**	**81.8**	**48.7**	**22.9**	**78.5**	**31.8**	**34.2**	0.0	**27.2**	**30.2**	**39.0**	**4.9**	43.9
FDA		14.5	0.9	23.2	1.0	**5.3**	1.1	7.6	0.9	28.4	0.0	57.9	3.0	0.2	8.2	3.8	0.0	0.0	0.0	1.6	8.3	42.1	50.4
FADA		17.4	7.6	18.1	1.2	2.1	0.4	0.5	0.1	29.2	0.0	11.8	3.8	0.2	18.5	0.0	0.1	0.0	0.0	0.0	5.8	43.4	49.2
IntraDA		26.4	3.0	46.3	0.4	4.5	0.7	8.6	0.5	30.9	0.4	43.9	21.3	1.2	47.5	8.3	7.5	0.0	0.2	6.5	13.6	32.7	46.3
CLAN		33.0	0.6	39.2	2.3	1.8	0.1	8.4	0.2	36.2	0.3	38.1	21.5	3.4	38.0	9.4	3.4	0.0	0.1	4.3	12.6	30.6	43.2
MaxSquare	0.5	17.0	0.3	33.6	0.6	2.2	0.4	9.9	0.4	29.5	0.0	31.2	3.5	0.4	28.8	5.7	0.4	0.0	0.0	1.3	8.7	37.7	46.4
AdaptSegNet		43.0	0.2	10.1	0.7	2.8	0.2	7.3	0.1	34.8	0.0	58.1	4.9	0.0	18.6	0.8	0.3	0.0	0.0	0.0	9.6	32.8	42.4
PCEDA		30.4	0.0	36.6	0.2	1.7	1.5	4.0	1.2	27.1	0.0	8.1	9.7	0.4	7.4	1.2	0.0	0.0	0.0	5.3	7.1	43.4	50.5
BDL		9.7	0.1	25.9	0.0	0.8	0.2	8.1	0.6	43.5	0.0	13.7	4.8	4.3	7.6	2.6	0.0	0.0	0.2	1.9	6.5	42.0	48.5
Ours		**82.9**	**10.0**	**49.8**	**3.4**	4.5	**12.7**	**20.7**	**19.9**	**59.9**	**5.8**	**78.6**	**35.9**	**12.6**	**60.2**	**18.9**	**18.2**	0.0	**10.8**	**15.5**	**27.4**	**16.5**	43.9

Table 4.2: Quantitative study of "GTA5 to Cityscapes". VGG16 (upper part) and ResNet101 (lower part) are used as backbones in this experiment. The performance is measured on 19 common classes with criteria: per-class IoU, mean IoU (mIoU), mIoU drop (performance degradation of the model after being attacked), and mIoU*. The higher the mIoU and the lower the mIoU drop, the more robust the model is. The best result in each column is highlighted in bold.

SYNTHIA to Cityscapes																				
	ϵ	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	bus	motorbike	bicycle	**mIoU**	**mIoU drop**	**mIoU***
FDA [39]	0.1	68.5	28.4	72.7	0.4	0.3	22.2	5.1	19.1	57.6	75.7	45.8	18.8	55.6	18.5	5.1	31.5	32.8	7.7	40.5
PCEDA [135]		80.9	25.0	**73.5**	**6.3**	**7.1**	14.2	**24.0**	**27.4**	76.2	70.3	45.0	19.9	70.0	20.3	**9.8**	25.1	37.2	3.9	41.1
BDL [34]		34.9	21.2	47.8	0.0	0.2	20.5	9.2	20.2	67.2	74.3	49.0	17.5	57.2	11.9	2.5	34.6	29.3	9.7	39.0
Ours		**88.2**	**46.5**	46.5	0.0	0.1	**24.6**	8.4	23.8	**79.3**	**81.2**	**54.4**	**24.5**	**78.2**	**22.4**	9.2	**44.4**	**41.3**	**-2.2**	39.1
FDA	0.25	46.3	16.0	38.7	0.0	0.2	4.9	2.5	8.9	31.3	38.9	8.6	5.3	17.7	6.0	1.3	5.4	14.5	26.0	40.5
PCEDA		75.6	11.4	59.1	0.0	**0.4**	9.6	5.5	12.9	63.1	45.0	30.7	13.4	34.9	8.6	2.5	24.5	24.8	16.3	41.1
BDL		8.0	8.9	31.1	0.0	0.1	8.7	6.9	9.8	52.0	54.1	22.9	4.9	25.6	2.5	0.8	13.3	13.6	25.4	39.0
Ours		**87.4**	**41.6**	**73.7**	0.0	0.1	**23.2**	**8.7**	**23.0**	**75.7**	**78.8**	**49.7**	**21.1**	**72.5**	**20.3**	**7.5**	**39.5**	**38.9**	**0.2**	39.1
FDA	0.5	42.2	4.9	14.2	0.0	0.1	0.6	1.0	1.7	26.2	1.9	0.5	0.4	1.5	0.1	0.1	0.1	6.0	34.5	40.5
PCEDA		66.2	1.1	47.9	0.0	**0.4**	3.1	2.5	5.0	47.8	18.8	10.0	1.9	8.3	3.2	1.1	10.2	14.2	26.9	41.1
BDL		0.6	1.0	24.8	0.0	0.0	1.6	1.9	2.3	35.8	18.6	2.2	0.1	4.1	0.1	0.0	0.5	5.9	33.1	39.0
Ours		**68.8**	**21.8**	**57.1**	0.0	0.1	**17.9**	**6.8**	**15.6**	**65.9**	**54.2**	**30.4**	**12.8**	**43.1**	**5.9**	**4.1**	**25.3**	**26.9**	**12.2**	39.1
FDA [39]	0.1	83.4	32.4	73.5	✗	✗	✗	13.1	**18.9**	71.6	79.5	**56.1**	24.9	77.5	27.6	18.2	**42.8**	47.7	4.8	52.5
FADA [98]		74.0	32.5	69.8	✗	✗	✗	6.8	15.8	57.0	58.3	46.7	8.6	55.1	18.0	4.5	9.8	35.1	17.4	52.5
DADA [54]		80.0	33.8	75.0	✗	✗	✗	8.0	9.4	62.1	76.3	49.7	14.3	76.3	27.8	5.2	31.7	42.3	7.5	49.8
MaxSquare [108]		70.1	23.3	72.8	✗	✗	✗	6.7	7.2	60.2	77.6	48.7	13.8	63.7	17.4	3.1	20.1	37.3	10.9	48.2
AdaptSegNet [19]		79.5	34.7	76.6	✗	✗	✗	4.1	5.4	61.0	80.8	49.3	18.3	72.1	26.1	7.5	29.8	41.9	4.8	46.7
PCEDA [135]		64.5	33.4	77.1	✗	✗	✗	**17.6**	16.5	50.1	81.3	48.9	24.8	71.9	25.7	13.3	41.0	43.6	10.0	53.6
BDL [34]		79.2	33.7	75.3	✗	✗	✗	5.6	8.7	61.1	80.6	45.0	21.7	65.7	26.7	8.5	24.5	41.2	10.2	51.4
Ours		**89.1**	**46.6**	**78.2**	✗	✗	✗	11.4	16.9	**76.1**	**81.5**	52.6	**26.7**	**79.9**	**35.3**	**25.0**	37.5	**50.5**	**-1.1**	49.4
FDA	0.25	8.6	9.0	40.8	✗	✗	✗	3.9	7.1	21.5	51.3	14.5	6.9	35.3	5.4	0.0	14.4	16.8	35.7	52.5
FADA		80.8	23.5	59.3	✗	✗	✗	1.7	3.7	50.6	15.6	26.2	0.8	21.2	6.2	0.3	2.1	22.5	30.0	52.5
DADA		58.0	11.5	42.7	✗	✗	✗	4.5	4.2	31.9	41.2	23.4	6.0	53.9	8.3	0.4	14.0	23.1	26.7	49.8
MaxSquare		70.3	4.6	53.1	✗	✗	✗	8.1	6.0	37.2	61.0	11.2	3.9	42.3	6.9	0.4	3.4	23.7	24.5	48.2
AdaptSegNet		28.4	7.6	56.8	✗	✗	✗	4.4	2.6	26.4	62.8	22.5	9.8	44.2	8.3	1.1	10.2	21.9	24.8	46.7
PCEDA		15.4	7.2	64.9	✗	✗	✗	9.3	9.8	27.0	71.4	35.3	13.9	52.0	12.3	2.2	25.4	26.7	26.9	53.6
BDL		46.9	9.1	65.5	✗	✗	✗	4.0	5.9	34.7	68.5	22.7	12.5	50.7	10.8	1.2	12.8	26.6	21.3	51.4
Ours		**87.4**	**25.0**	**70.7**	✗	✗	✗	**10.9**	**18.2**	**60.0**	**74.9**	**43.8**	**20.7**	**64.8**	**17.7**	**4.5**	**29.9**	**40.7**	**8.7**	49.4
FDA	0.5	0.0	0.0	7.2	✗	✗	✗	1.3	0.7	17.8	13.7	0.0	0.0	2.5	0.2	0.0	0.0	3.3	49.2	52.5
FADA		**76.0**	**15.9**	**56.3**	✗	✗	✗	0.2	0.6	**45.0**	0.2	7.6	0.0	5.2	0.9	0.0	0.1	16.0	36.5	52.5
DADA		42.9	2.3	16.3	✗	✗	✗	1.8	0.7	24.1	12.5	2.5	0.8	23.5	2.1	0.0	4.8	10.3	39.5	49.8
MaxSquare		42.7	0.2	25.3	✗	✗	✗	5.0	2.7	24.5	18.0	0.8	0.1	15.0	1.5	0.0	0.2	10.5	37.7	48.2
AdaptSegNet		2.1	0.4	24.5	✗	✗	✗	2.1	0.5	19.2	21.4	1.4	2.2	11.7	1.7	0.1	2.5	6.9	39.8	46.7
PCEDA		0.1	0.1	40.0	✗	✗	✗	2.4	1.8	21.0	37.2	**13.1**	1.3	9.3	2.5	0.7	1.6	10.1	43.5	53.6
BDL		2.8	0.7	32.1	✗	✗	✗	2.0	1.8	20.3	53.7	2.7	1.3	22.3	1.4	0.4	1.7	11.0	40.4	51.4
Ours		65.5	4.3	44.0	✗	✗	✗	**6.6**	**13.7**	31.9	**60.8**	12.6	**7.8**	**24.8**	**3.4**	**1.2**	**14.4**	**22.4**	**27.0**	49.4

Table 4.3: Quantitative study of "SYNTHIA to Cityscapes". VGG16 (upper part) and ResNet101 (lower part) are used as backbones in this experiment. The comparison is performed on 16 common classes for VGG16 and 13 common classes for ResNet101.

the tradition to use the training images from Cityscapes as the target domain and use the validation images as the clean test data.

4.4.2 Implementation Details

Following the same experimental protocol in this area, we use two network architectures: DeepLab-v2 [70] with VGG16 [86] backbone, and DeepLab-v2 with ResNet101 backbone. The domain discriminator has 5 convolution layers with kernel 4×4 and stride of 2, each of which is followed by a leaky ReLU parameterized by 0.2

except the last one. The channel number of each layer is {64, 128, 256, 512, 1}. The Adam optimizer with initial learning rate 1e-4 and momentum (0.9, 0.99) is used in DeepLab-VGG16. We apply step decay to the learning rate with step size 30000 and drop factor 0.1. Stochastic Gradient Descent optimizer with momentum 0.9 and weight decay 5e-4 is used in DeepLab-ResNet101. The learning rate of DeepLab-ResNet101 is initialized as 1e-4 and is decreased by the polynomial policy with a power of 0.9. Adam optimizer with momentum (0.9, 0.99) and initial learning rate 1e-6 is used in the domain discriminator. We set $\epsilon_m = 1.0$ in equation 4.5. Code and data are available at https://github.com/uta-smile/ASSUDA.

4.4.3 Perturbed Test Data

To evaluate model robustness, we first generate the perturbed test data. Specifically, PSPNet [69] is used as the surrogate model owing to its popularity. We generate three sets of perturbed test data using FGSM with $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$. The generated perturbed data sets are then used for performance assessment. For a fair comparison with existing UDA methods, we download the pre-trained models from the original papers and perform the evaluation.

4.4.4 Experimental Results

Since the robustness of existing UDA methods remains unexplored, we first comprehensively evaluate their robustness against adversarial attacks in this section (Table 4.2 and Table 4.3). We then perform a comparison of our method on two widely used benchmark settings, *i.e.*, "GTA5 to Cityscapes" and "SYNTHIA to Cityscapes". Three criteria, *i.e.*, mIoU, mIoU drop, and mIoU* are used for performance assessment. Specifically, mIoU and mIoU* indicate the mean IoU on the perturbed test data and the clean test data, respectively, while mIoU drop indicates the performance

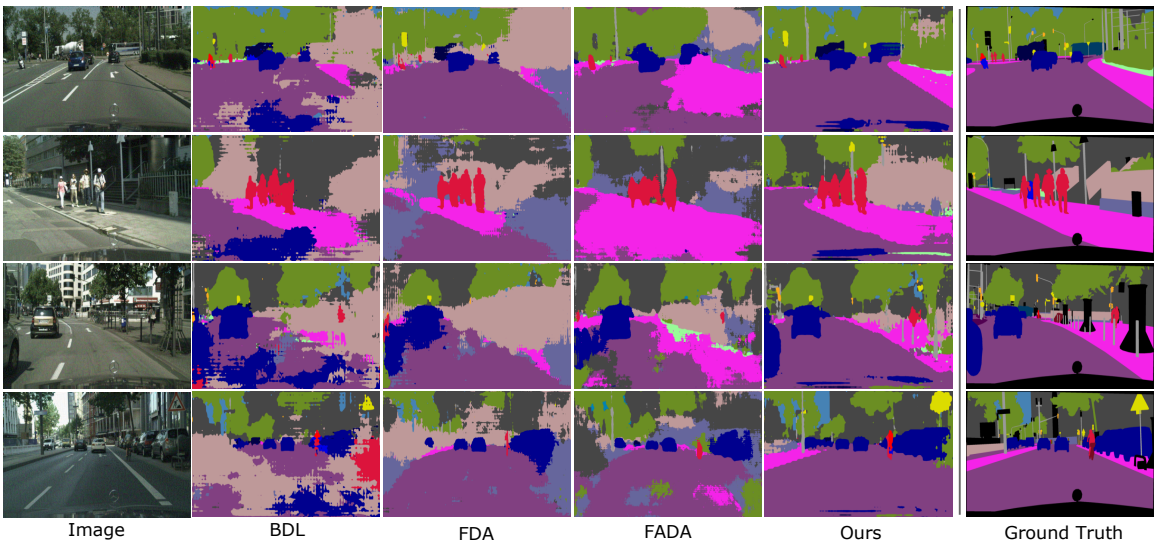

Figure 4.3: Qualitative comparison of our method against BDL [34], FDA [39], and FADA [98] on the perturbed test data ($\epsilon = 0.25$). All of these models are trained on "GTA5 to Cityscapes" with ResNet101. The first column indicates perturbed test images.

degradation (*i.e.*, the difference between mIoU and mIoU*). Therefore, the higher the mIoU and the lower the mIoU drop, the more robust the model is.

4.4.4.1 GTA5 to Cityscapes

As shown in Table 4.2, we achieve the best performance on all three adversarial attacks. In particular, even slight adversarial perturbations can mislead AdaptSegNet [19] and BDL [34] and dramatically degrade their performance. For instance, when evaluated with VGG16 backbone on perturbed test data from $\epsilon = 0.25$, they only achieve mIoU 7.9 and mIoU 14.6, with mIoU drop 27.1 and 26.7, respectively. Similarly, two recently proposed UDA methods, *i.e.*, FDA [39] and PCEDA [135] suffer from mIoU drop of 31.4 and 30.8, respectively. By contrast, our method still gets mIoU 35.2 and only has a performance drop of mIoU 4.5. The results suggest that existing UDA methods in semantic segmentation are broadly vulnerable to adversarial attacks.

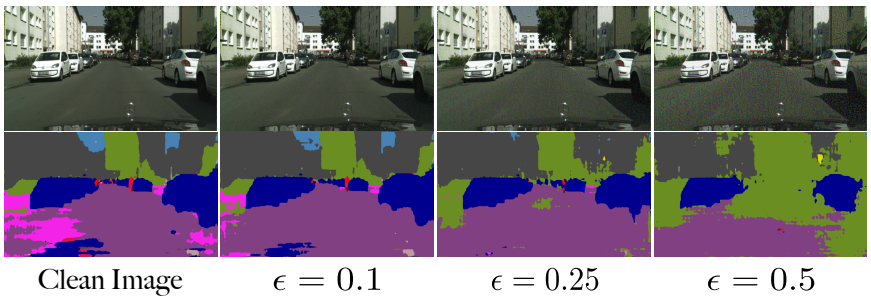

Figure 4.4: Qualitative study of our method under three adversarial attacks, *i.e.*, $\epsilon = 0.1$, $\epsilon = 0.25$, and $\epsilon = 0.5$. All of these models are trained on "SYNTHIA to Cityscapes" with ResNet101.

The reason is that although these methods demonstrate remarkable performance on the clean test data (as indicated by mIoU*), none of them, however, take the adversarial attack into account during learning transferable representations. Instead, we innovatively propose adversarial self-supervision to improve the robustness of UDA models by taking advantage of both adversarial training and self-supervision. This is evidenced by the qualitative study in Figure 4.3, where our method demonstrates accurate predictions on the perturbed test data.

In terms of the clean performance (or mIoU*), our method usually lags behind the existing state of the arts. This is consistent with recent studies that clean performance and adversarial robustness might be at odds [136, 137].

4.4.4.2 SYNTHIA to Cityscapes

Table 4.3 shows the performance comparison on "SYNTHIA to Cityscapes", where our method again demonstrates significant robustness improvement. In contrast, other UDA methods can be easily fooled by small perturbations in the test data. Interestingly, our method achieves better performance on the perturbed test data ($\epsilon = 0.1$) than on the clean test data. This can be explained by the fact that training on adversarial examples can regularize the model somewhat, as reported in [130, 129].

	GTA5 to Cityscapes		SYNTHIA to Cityscapes	
ϵ	$\delta = 0$	Ours	$\delta = 0$	Ours
0.1	39.2	39.3	41.5	41.3
0.25	33.8	35.2	36.7	38.9
0.5	21.8	24.8	23.6	26.9
0.1	43.3	43.3	49.7	50.5
0.25	37.8	39.0	37.8	40.7
0.5	24.3	27.4	15.7	22.4

Table 4.4: Ablation study of δ with backbone VGG16 (upper part) and ResNet101 (lower part).

We further perform a qualitative study of our method when evaluated on the test data with different magnitudes of the perturbation. As shown in Figure 4.4, although large ϵ usually results in worse performance, our method still demonstrates robust predictions.

4.4.4.3 Ablation Study

To learn the contribution of the self-supervision, we conduct the ablation study in Table 4.4. Compared to $\delta = 0$ which only contains self-training, incorporating self-supervision consistently improves the performance. We further investigate the training perturbation magnitude ϵ_m in equation 4.5. Table 4.5 reveals that $\epsilon_m = 1.0$ (Ours) results in more robust UDA model than $\epsilon_m = 0.1$. The reason is that the adversarial examples generated by $\epsilon_m = 1.0$ are highly perturbed compared to the adversarial examples from $\epsilon_m = 0.1$, which in turn encourages our model to be more robust against perturbations.

	VGG16		**ResNet101**	
ϵ	$\epsilon_m = 0.1$	$\epsilon_m = 1.0$	$\epsilon_m = 0.1$	$\epsilon_m = 1.0$
0.1	36.4	39.3	44.9	43.3
0.25	17.8	35.2	34.3	39.0
0.5	7.4	24.8	15.7	27.4

Table 4.5: Ablation study of ϵ_m on "GTA5 to Cityscapes".

4.5 Summary and Discussion

In this chapter, we introduce a new unsupervised domain adaptation framework for semantic segmentation. This is motivated by the observation that the robustness of semantic adaptation methods against adversarial attacks has not been investigated. Our pilot studies reveal that existing UDA methods can be easily deceived by unnoticeable perturbations. We therefore propose adversarial self-supervision by maximizing agreement between clean samples and their adversarial examples to improve model robustness. Extensive empirical studies are performed to explore the benefits of our method in improving the model robustness against adversarial attacks. The effectiveness of our method is thoroughly proved on commonly used benchmarks.

It is noteworthy that existing UDA studies are mainly built upon convolutional neural networks (CNNs) to learn domain-invariant representations. With the recent exponential increase in applying Vision Transformer (ViT) [40] to vision tasks, the capability of ViT in adapting cross-domain knowledge, however, remains unexplored in the literature. This will be our main focus in the next chapter.

CHAPTER 5

TVT: Transferable Vision Transformer for Unsupervised Domain Adaptation

In this chapter, we first comprehensively investigates the performance of ViT [40] on a variety of domain adaptation tasks. Surprisingly, ViT demonstrates superior generalization ability, while the performance can be further improved by incorporating adversarial adaptation. Notwithstanding, directly using CNNs-based adaptation strategies fails to take the advantage of ViT's intrinsic merits (e.g., attention mechanism and sequential image representation) which play an important role in knowledge transfer. To remedy this, we propose an unified framework, namely Transferable Vision Transformer (TVT), to fully exploit the transferability of ViT for domain adaptation. Specifically, we delicately devise a novel and effective unit, which we term Transferability Adaption Module (TAM). By injecting learned transferabilities into attention blocks, TAM compels ViT focus on both transferable and discriminative features. Besides, we leverage discriminative clustering to enhance feature diversity and separation which are undermined during adversarial domain alignment. To verify its versatility, we perform extensive studies of TVT on four benchmarks and the experimental results demonstrate that TVT attains significant improvements compared to existing state-of-the-art UDA methods.

5.1 Introduction

Deep neural networks (DNNs) demonstrate unprecedented achievements on various machine learning problems and applications. However, such impressive performance heavily relies on massive amounts of labeled data which requires considerable

time and labor efforts to collect. Therefore, it is desirable to train models that can leverage rich labeled data from a different but related domain and generalize well on target domains with no or limited labeled examples. Unfortunately, the canonical supervised-learning paradigm suffers from the domain shift issue that poses a major challenge in adapting models across domains. This motivates the research on unsupervised domain adaptation (UDA) [138] which is a special scenario of transfer learning [8]. The key idea of UDA is to project data points of the labeled source domain and the unlabeled target domain into a common feature space, such that the projected features are both discriminative (semantic meaningful) and domain-invariant, in turn, generalize well to bridge the domain gap. To achieve this goal, various methods have been proposed in the past decades, among which adversarial adaptation has become the dominant technique in this field, which attempts to align cross-domain representations by minimizing an adversarial loss through a domain discriminator [24, 25, 31].

Recently, Vision Transformer (ViT) [40] has received increasing attention in the vision community. Different from CNNs that act on local receptive fields of the given image, ViT models long-range dependencies among visual features across the entire image, through the global self-attention mechanism. Specifically in ViT, each image is split into a sequence of fixed-size non-overlapping patches, which are then linearly embedded and concatenated with position embeddings. To be consistent with NLP paradigm, a class token is prepended to the patch tokens, serving as the representation of the whole image. Then, those sequential embeddings are fed into a stack of transformers to learn desired visual representations. Due to its advantages in global context modeling, ViT has obtained excellent results on various vision tasks, such as image classification [40], object detection [139, 140], segmentation [141, 45], and video understanding [142, 143].

Despite that ViT is becoming increasingly popular, two important questions related to domain adaption remain unanswered. First, *how does the generalization ability of ViT across different domains?* There are several contemporary work [41, 42, 43] that apply DeiT [44] and Swin [45] to UDA, yet the ViT has not been investigated. The second question is, *how can we properly improve ViT in adapting different domains?* One intuitive approach is to directly apply adversarial discriminator onto the class tokens to perform adversarial alignment, where the state of a class token represents the entire image. However, cross-domain alignment of such global features assumes all regions or aspects of the image have the equal transferability and discriminative potential, which is not always tenable. For instance, background regions can be easier aligned across domains, while foreground regions are more discriminative. In other words, some discriminative features may lack transferability, and some transferable features may not contribute much to the downstream task (e.g., classification). Therefore, in order to properly enhance the transferability of ViT, it is essential to identify fine-grained features that are both transferable and discriminative.

In this chapter we aim to present our answers to the two aforementioned questions. Firstly, to fill the blank of understanding ViT's generalization ability, we first conduct a comprehensive study of vanilla ViT [40] on public UDA benchmarks. As expected, our experimental results demonstrate that ViT even in the source-only setting outperforms its strong CNNs-based counterparts. There could be multiple deep reasons behind the strong performance of ViT [144, 145], which are not in the scope of this chapter. Besides, we observe further improvements by applying an adversarial discriminator to the class tokens of ViT, which only aligns global representations. However, such strategy suffers from the oversimplified assumption and ignores the inherent properties of ViT that are beneficial for domain adaptation: i) sequential patch tokens actually give us the free access to fine-grained features; ii) the self-attention mechanism in

transformer naturally works as a discriminative probe. In the light of this, we propose an unified UDA framework that makes full use of ViT's inherent merits. We name it Transferable Vision Transformer (TVT).

The key idea of our method is to retain both transferable and discriminative features which are essential in knowledge adaptation. To achieve this goal, we first introduce the novel Transferability Adaption Module (TAM) built upon a conventional transformer. TAM uses a patch-level domain discriminator to measure the transferabilities of patch tokens, and injects learned transferabilities into the multi-head self-attention block of a transformer. On one hand, the attention weights of patch tokens in the self-attention block are used to determine their semantic importance, i.e., the features with larger attention are more discriminative yet without transferability guarantees. On the other hand, as patch tokens can be regarded as fine-grained representations of an image, the higher transferability of a token means the local features are more transferable across domains though not necessarily discriminative. By simply replacing the last transformer of ViT with a plug-and-play TAM, we could drive ViT to focus on both transferable and discriminative features.

Since our method performs adversarial adaptation that forces the learned features of two domains to be similar, one underlying side-effect is that the discriminative information of target domain might be destroyed during feature alignment. To address this problem, we design a Discriminative Clustering Module (DCM) inspired by the clustering assumption. The motivation is to enforce the individual target prediction close to one-hot encoding (well separated) and the global target prediction to be uniformly distributed (global diverse), such that the learnt target-domain representation could retain maximum discriminative information about the input values.

Contributions of this chapter are summarized as follows:

- As far as we know, we are the first investigating the capability of ViT in transferring knowledge on the domain adaptation task. We believe this work gives good insights to understand and explore ViT's generalization ability while applied to various vision tasks.
- We propose TAM that delicately leverages the intrinsic characteristics of ViT, such that our method can capture both transferable and discriminative features for domain adaptation. Moreover, we adopt discriminative clustering assumption to alleviate the discrimination destruction during adversarial alignment.
- Without any bells and whistles, our method set up a new competitive baseline cross several public UDA benchmarks.

5.2 Related Work

5.2.1 Unsupervised Domain Adaptation

Transfer learning aims to learn transferable knowledge that are generalizable across different domains with different distributions [8, 76]. This is built upon the evidence that feature representations in machine learning models, especially in deep neural networks, are transferable [146]. The main challenge of transfer learning is to reduce the domain shift or the discrepancy of the marginal probability distributions across domains [138]. In the past decades, various methods have been proposed to address one canonical transfer learning problem, i.e., unsupervised domain adaptation (UDA), where no labels are available for the target domain. For instance, DDC [12] attempted to learn domain-invariant features by minimizing Maximum Mean Discrepancy (MMD) [29] between two domains. Long et al. further improved DDC by embedding hidden representations of all task-specific layers in a reproducing Hilbert space and used a multiple kernel variant of MMD to measure the domain distance [13].

Long et al. proposed to align joint distributions of multiple domain-specific layers across domains through a joint maximum mean discrepancy metric [28]. Another line of effort was inspired by the success of adversarial learning [30]. By introducing a domain discriminator and modeling the domain adaption as a minimax problem [24, 25, 31], an encoder is trained to generate domain-invariant features, through deceiving a discriminator which tries to distinguish features of source domain from that of target domain.

It is noteworthy that all of these methods completely or partially used CNNs as the fundamental block [26, 2, 7]. By contrast, our method explores ViT [40] to tackle the UDA problem, as we believe ViT has better potential and capability in domain adaptation owning to some of its properties. Although previous UDA methods (e.g., adversarial learning) are able to improve vanilla ViT to some extent, they were not well designed for transformer-based models, and thereby cannot leverage ViT's inherent characteristic of providing attention information and fine-grained representations. However, Our method is delicately designed with the nature of ViT and could effectively leverages the transferability and discrimination of each feature for knowledge transfer, thus having better chance in fully exploiting the adaptation power of ViT.

5.2.2 Vision Transformer

Transformers [101] was firstly proposed in the NLP field and demonstrate record-breaking performance on various language tasks, e.g., text classification and machine translation [147, 148, 149]. Much of such impressive achievement is attributed to the power of capturing long-range dependencies through attention mechanism. Spurred by this, some recent studies attempted to integrate attention into CNNs to augment feature maps, aiming to provide the capability in modeling heterogeneous interactions

[150, 151, 152]. Another pioneering work of completely convolution-free architecture is Vision Transformer (ViT), which applied transformers on a sequence of fixed-size non-overlapping image patches. Different from CNNs that rely on image-specific inductive biases (e.g., locality and translation equivariance), ViT takes the benefits from large-scale pre-training data and global context modeling. One such method [40], known for its simplicity and accuracy/compute trade-off, competes favorably against CNNs on the classification task and lays the foundation for applying transformer to different vision tasks. ViT and its variants have proved their wide applicability in object detection [139, 153, 140], segmentation [141, 154], and video understanding [142, 143], etc.

Despite the success of ViT on different vision tasks, to the best of our knowledge, neither their transferability nor the design of UDA methods with ViT have been previously discussed in the literature. To this end, we focus in this chapter on the investigation of ViT's capability in knowledge transferring across different domains. Furthermore, we propose a novel UDA framework tailored for ViT by exploring its intrinsic merits and prove its superiority over existing methods. It is noteworthy that there are several contemporary work [41, 42, 43] that apply DeiT [44] and Swin [45] to UDA. Specifically, [41, 42] uses cross-attention to obtain the mixup representations of source and target images, [43] uses two class tokens to learn domain-specific information. Different from these works, our study focuses on the empirical investigation of ViT's generalization ability and proposes a plug-and-play module to boost ViT's performance in knowledge transfer.

5.3 Preliminaries

5.3.1 Adversarial Learning UDA

We consider the image classification task in UDA, where a labeled source domain $\mathcal{D}_s\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s examples and an unlabeled target domain $\mathcal{D}_t\{x_j^t\}_{j=1}^{n_t}$ with n_t examples are given. The goal of UDA is to learn features that are both discriminative and invariant to the domain discrepancy, and in turn guarantee accurate prediction on the unlabeled target data. Here, a common practice is to jointly performs feature learning, domain adaptation, and classifier learning by optimizing the following loss function:

$$\mathcal{L}_{clc}(x^s, y^s) + \alpha\mathcal{L}_{dis}(x^s, x^t) \tag{5.1}$$

where $\mathcal{L}_{clc}$ is supervised classification loss, $\mathcal{L}_{dis}$ is a transfer loss with various possible implementations, and α is used to control the importance of $\mathcal{L}_{dis}$. One of the most commonly used $\mathcal{L}_{dis}$ is the adversarial loss which encourages a domain-invariant feature space through a domain discriminator [24].

5.3.2 Self-attention Mechanism

The main building block of ViT is Multi-head Self-Attention (MSA), which is used in the transformer to capture long-range dependencies [101]. Specifically, MSA concatenates multiple scaled dot-product attention (short for SA) modules, where each SA module takes a set of queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$) as inputs. In order to learn dependencies between distinct positions, SA computes the dot products of the query with all keys, and applies a softmax function to obtain the weights on the values.

$$\mathrm{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V} \tag{5.2}$$

where d is the dimension of $\mathbf{Q}$ and $\mathbf{K}$. With $\mathrm{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, MSA is defined as:

$$\begin{aligned} \mathrm{MSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_k)\mathbf{W}^O \\ \text{where } \mathrm{head}_i &= \mathrm{SA}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \end{aligned} \tag{5.3}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ are projections of different heads, $\mathbf{W}^O$ is another mapping function. Intuitively, using multiple heads allows MSA to jointly attend to information from different representation subspaces at different positions.

5.4 Methodology

In this section, we first investigate ViT's ability in knowledge transfer on various adaptation tasks. After that, we conduct the early attempts to improve ViT's transferability by incorporating adversarial learning. Finally, we introduce our method named Transferable Vision Transformer (TVT), which consists two new adaptation modules to further improve ViT's capability for cross-domain adaptation..

5.4.1 ViT's Generalization Ability

To the best of our knowledge, the generalization ability of ViT has not been studied in the literature before, although ViT and its variants have shown great success in various vision task. To probe into ViT's capability of domain adaptation, we choose the vanilla ViT [40] as the backbone in all of our studies, owing to its simplicity and popularity. We train vanilla ViT by labeled source data only and assess its generalization ability by the classification accuracy on target data. As mentioned above, CNNs-based approaches dominate UDA research in the past decades and demonstrate great successes. Therefore, we compare vanilla ViT with CNNs-based architectures, including LeNet [26], AlexNet [2], and ResNet [7]. All experiments are performed on well-established benchmarks with standard evaluation protocols.

Take the results on Office-31 dataset for example. As shown in Table 5.2, Source Only ViT obtains impressing classification accuracy 89.5%, which is much better than its strong CNN opponents AlexNet (70.1%) and ResNet (76.1%). Similar phenomenon can be observed in other benchmark results, where ViT competes favorably against, if not better than, the other state-of-the-arts CNNs backbones, as shown in Table 5.1,5.3,5.4. Surprisingly, Source Only ViT even outperforms strong CNNs-based UDA approaches without any bells and whistles. For instance, it achieves an average accuracy 78.7% on Office-Home dataset (Table 5.3), beating all CNN-based UDA methods. Compared to SHOT [155] recognized as the best UDA model nowadays, Source Only ViT obtains 7% absolute accuracy boost, a big step in pushing the frontier of UDA research. There could be multiple reasons behind the strong performance of ViT [144, 145], for example, the striking differences between the features learned by ViTs and CNNs [144]. We leave this as future work. Despite this, a large gap still exists between the Source Only and Target Only models (88.3% vs 99.2%) as shown in Table 5.1, which indicates potential improvement space of ViT's generalization ability.

5.4.2 ViT w/ Adversarial Adaptation: Baseline

We first investigate how ViT benefits from adversarial adaptation [24], which is widely used in CNNs-based UDA methods. We follow the typical adversarial adaptation fashion that employs an encoder G_f for feature learning, a classifier G_c for classification, and a domain discriminator D_g for global feature alignment. Here, G_f is implemented as ViT and D_g is applied to output state of the class tokens of the source and target images. To accomplish domain knowledge adaptation, G_f and D_g play a minimax game: G_f learns domain-invariant features to deceive D_g, while D_g

distinguishes source-domain features from that of target-domain. The objective can be formulated as:

$$\begin{aligned}\mathcal{L}_{clc}(x^s, y^s) &= \frac{1}{n_s}\sum_{x_i \in \mathcal{D}_s} \mathcal{L}_{ce}(G_c(G_f(x_i^s)), y_i^s) \\ \mathcal{L}_{dis}(x^s, x^t) &= -\frac{1}{n}\sum_{x_i \in \mathcal{D}} \mathcal{L}_{ce}(D_g(G_f(x_i^*)), y_i^d),\end{aligned} \tag{5.4}$$

where $n = n_s + n_t$, $\mathcal{D} = \mathcal{D}_s \bigcup \mathcal{D}_t$, $\mathcal{L}_{ce}$ is cross-entropy loss, the superscript $*$ can be either s or t to denote a source or a target domain, and y^d denotes the domain label (i.e., $y^d = 1$ is source, $y^d = 0$ is target).

We denote ViT with adversarial adaptation as our Baseline. As shown in Table 5.1,5.2,5.3,5.4, Baseline shows 7.8%, 0.8%, 1.6%, and 3.2% absolute accuracy improvements over vanilla ViT, respectively on the four benchmarks. Those results reveal that global feature alignment with a domain discriminator helps ViT's generalization ability. However, compared with the digit recognition task, Baseline achieves limited improvements on object detection which is more complicated and challenging. We boils down such observation to a conclusion that simply applying global adversarial alignment cannot exploit ViT's full transferable power, since it fails to consider two key factors: (i) not all regions/features are equally transferable or discriminative. For effective knowledge transfer, it is essential to focus on both transferable and discriminative features; (ii) ViT naturally provides fine-grained features given its forward passing sequential tokens, and attention weights in transformer actually convey discriminative potentials of patch tokens. To address these challenges and fully leverage the merits of ViT, a new UDA framework named Transferable Vision Transformer (TVT) is further proposed.

5.4.3 Transferable Vision Transformer (TVT)

An overview of TVT is shown in Figure 5.1, which contains two main modules: (i) a Transferability Adaptation Module (TAM) and (ii) a Discriminative Clustering Module (DCM). These two modules are highly interrelated and play a complementary role in transferring knowledge for ViT-based architectures. TAM encourages the output state of class token to focus on both transferable and semantic meaningful features, and DCM enforces the aligned features of target-domain samples to be clustered with large margins. As a consequence, the features learnt by TVT are discriminative in classification and transferable across domains as well. We detail each module in what follows.

5.4.4 Transferability Adaptation Module

As shown in Figure 5.1, we introduce the Transferability Adaptation Module (TAM) that explicitly considers the intrinsic merits of ViT, i.e., attention mechanisms and sequential patch tokens.

As the patch tokens are regarded as local features of an image, they are corresponded to different image regions or captures different visual aspects as fine-grained representations of an image. Assuming patch tokens of different semantic importance and transferabilities, TAM aims at assigning different weights to those tokens, to encourage the learned image representations, i.e., the output state of class token, to attend to patch tokens that are both transferable and discriminative. While the self-attention weights in ViT could be employed as discriminative weights, one major hurdle here is, the transferability of each patch token is not available. To bypass this

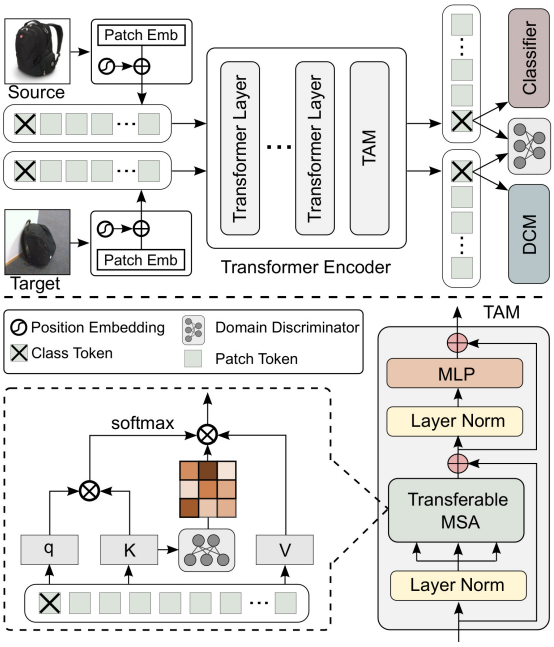

Figure 5.1: An overview of the proposed TVT framework. As in ViT, both source and target images are split into fixed-size patches which are then linearly mapped and embedded with positional information. The generated patches are fed into a transformer encoder whose last layer is replaced by Transferability Adaptation Module (TAM). Feature learning, adversarial domain adaptation and classification are accomplished by ViT-akin backbone, two domain discriminators (on patch-level and global-level), Discriminative Clustering Module (DCM) and the MLP-based classifier

difficulty, we adopt a patch-level domain discriminator D_l that matches cross-domain local features [156, 157] by optimizing:

$$\mathcal{L}_{pat}(x^s, x^t) = -\frac{1}{nR} \sum_{x_i \in \mathcal{D}} \sum_{r=1}^{R} \mathcal{L}_{ce}(D_l(G_f(x_{ir}^*)), y_{ir}^d), \tag{5.5}$$

where R is number of patches, and $D_l(f_{ir})$ is the probability of this region belonging to the source domain. During adversarial learning, D_l tries to assign 1 for a source-domain patch and 0 for the target-domain ones, while G_f combats such circumstances. Conceptually, a patch that can easily deceive D_l (i.e., D_l is around 0.5) is more transferable across domains and should be given a higher transferability. We therefore use $t_{ir} = T(f_{ir}) = H(D_l(f_{ir})) \in [0, 1]$ to measure the transferability of r^{th} token of

i^{th} image, where $H(\cdot)$ is the standard entropy function. An other explanation of the transferability is: by assigning weights to different patches, it disentangles an image into common space representations and domain-specific representations, while the passing paths of domain-specific features are softly suppressed.

We then convert the conventional MSA into the transferable MSA (T-MSA) by transferability adaptation, i.e., injecting the learned transferabilities into attention weights of the class token. Our T-MSA is built upon the transferable self-attention (TSA) block that is formally defined as:

$$\text{TSA}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{q}\mathbf{K}^T}{\sqrt{d}}) \odot [1; T(\mathbf{K}_{patch})]\mathbf{V} \tag{5.6}$$

where $\mathbf{q}$ is the query of the class token, $\mathbf{K}_{patch}$ is the key of the patch tokens, $\odot$ is Hadamard product, and $[;]$ is concatenation operation. Obviously, $\text{softmax}(\frac{\mathbf{q}\mathbf{K}^T}{\sqrt{d}})$ and $[1; T(\mathbf{K}_{patch})]$ indicate the discrimination (semantic importance) and the transferability of each patch token, respectively. To jointly attend to the transferabilities of different representation subspaces and of different locations, we thus define T-MSA as:

$$\begin{aligned} \text{T-MSA}(\mathbf{q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}(\text{head}_1, ..., \text{head}_k)\mathbf{W}^O \\ \text{where head}_i &= \text{TSA}(\mathbf{q}\mathbf{W}_i^q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \end{aligned} \tag{5.7}$$

Taken them together, we get the TAM as follows:

$$\begin{aligned} \hat{\mathbf{z}}^l &= \text{T-MSA}(\text{LN}(\mathbf{z}^{l-1})) + \mathbf{z}^{l-1} \\ \mathbf{z}^l &= \text{MLP}(\text{LN}(\hat{\mathbf{z}}^l)) + \hat{\mathbf{z}}^l, \end{aligned} \tag{5.8}$$

where LN is LayerNorm layer, MLP denotes Multi-Layer Perception, $\mathbf{z}^l$ is hidden representation at layer l. We only apply TAM to the last transformer layer where patch features are spatially non-local and of higher semantic meanings. By this means, TAM focuses on fine-grained features that are transferable across domains and are discriminative for classification. So we have $l = L$, where L is the total number of transformer layers in ViT.

5.4.5 Discriminative Clustering Module

Towards the challenging problem of learning a probabilistic discriminative classifier with unlabeled target data, it is desirable to minimize the expected classification error on the target domain. However, cross-domain feature alignment through TAM by forcing the two domains to be similar may destroy the discriminative information of the learned representation, if no semantic constrains of the target domain is introduced. As shown in Figure 5.2, although the target feature is indistinguishable from the source feature, it is distributed in a mess which limits its discriminative power. To address this limitation, we are inspired by the assumptions that: (i) $p^t = \text{softmax}(G_c(G_f(x^t)))$ are expected to retain as much information about x^t as possible [158]; and (ii) decision boundary should not cross high density regions, but instead lie in low density regions, which is also known as cluster assumption [159]. Fortunately, these two assumptions can be met by maximizing mutual information between the empirical distribution on the target inputs and the induced target label distribution [160, 161, 162], which can be formally defined as:

$$\begin{aligned} \mathcal{I}(p^t; x^t) &= H(\bar{p^t}) - \frac{1}{n_t}\sum_{j=1}^{n_t} H(p_j^t) \\ &= -\sum_{k=1}^{K} \bar{p_k^t} log(\bar{p_k^t}) + \frac{1}{n_t}\sum_{j=1}^{n_t}\sum_{k=1}^{K} p_{jk}^t log(p_{jk}^t) \end{aligned} \tag{5.9}$$

where $p_j^t = \text{softmax}(G_c(G_f(x_j^t)))$, $\bar{p^t} = \mathbb{E}_{x_t}[p^t]$, and K is the number of classes. Note that maximizing $-\frac{1}{n_t}\sum_{j=1}^{n_t} H(p_j^t)$ enforces the target predictions close to one-hot encoding, therefore the cluster assumption is guaranteed. To ensure the global diversity, we also maximize $H(\bar{p^t})$ to avoid that every target data is assigned to the same class. With $\mathcal{I}(p^t; x^t)$, our model is encouraged to learn tightly clustered target features with uniform distribution, such that the discriminative information in the target domain are retained.

Table 5.1: Performance comparison on the Digits dataset. TVT* indicates that the backbone is pre-trained on ImageNet

Algorithm		S→M	U→M	M→U	Avg
Source Only	LeNet	67.1	69.6	82.2	73.0
RevGrad [163]	LeNet	73.9	73.0	77.1	74.7
ADDA [25]	LeNet	76.0	90.1	89.4	85.2
SHOT-IM [155]	LeNet	89.6	96.8	91.9	92.8
CyCADA [164]	LeNet	90.4	96.5	95.6	94.2
CDAN [31]	LeNet	89.2	98.0	95.6	94.3
MCD [165]	LeNet	96.2	94.1	94.2	94.8
Target Only		99.4	99.4	98.0	98.9
Source Only	ViT	88.6	88.2	73.1	88.3
Baseline	ViT	92.7	98.6	97.0	96.1
TVT*	ViT	98.0	98.9	97.7	98.2
TVT	ViT	99.0	99.4	98.2	98.9
Target Only		99.7	99.7	98.3	99.2

To summarize, the objective function of TVT is:

$$\mathcal{L}_{clc}(x^s, y^s) + \alpha\mathcal{L}_{dis}(x^s, x^t) + \beta\mathcal{L}_{pat}(x^s, x^t) - \gamma\mathcal{I}(p^t; x^t) \tag{5.10}$$

where α, β, and γ are hyper-parameters.

5.5 Experiments

To verify the effectiveness of our model, we conduct comprehensive studies on commonly used benchmarks and present experimental comparisons against state-of-the-art UDA methods as shown below.

5.5.1 Datasets

5.5.1.1 Digits

is an UDA benchmark on digit classification. We follow the same setting in previous work to perform adaptations on MNIST [26], USPS, and Street View House Numbers (SVHN) [27]. For each source-target domain pair, we train our model using

the training sets of each domain, and perform evaluations on the standard test set of the target domain.

5.5.1.2 Office-31

[166] contains 4,652 images of 31 categories, which were collected from three domains: Amazon (A), DSLR (D), and Webcam (W). The Amazon (A) image were downloaded from amazon.zom, while the DSLR (D), and Webcam (W) were photoed under the office environment by web and digital SLR camera, respectively.

5.5.1.3 Office-Home

[167] consists of images from four different domains: Artistic images (Ar), Clip Art (Cl), Product im- ages (Pr), and Real-World images (Rw). A total of 65 categories are covered within each domain.

5.5.1.4 VisDA-2017

[171] is a synthesis-to-real object recognition task used for the 2018 VisDA challenge. It covers 12 categories. The source domain contains 152,397 synthetic 2D renderings generated from different angles and under different lighting conditions, while the target domain contains 55,388 real-world images.

5.5.2 Existing Methods

We use the results in their original papers for fair comparison. For each type of backbone, we report its lower bound performance, denoted as Source Only, meaning the models are trained with source data only. For digit recognition, we also show the Target Only results as the high-end performance, which is obtained by both training

Table 5.2: Performance comparison on the Office-31 dataset. TVT* indicates that the backbone is pre-trained on ImageNet. "-S" and "-B" indicate that the backbone is DeiT-Small and DeiT-Base, respectively

Algorithm		A→ W	D→ W	W→ D	A→ D	D→ A	W→ A	Avg
Source Only	AlexNet	61.6	95.4	99.0	63.8	51.1	49.8	70.1
DDC [12]	AlexNet	61.8	95.0	98.5	64.4	52.1	52.2	70.6
DAN [13]	AlexNet	68.5	96.0	99.0	67.0	54.0	53.1	72.9
RevGrad [163]	AlexNet	73.0	96.4	99.2	72.3	53.4	51.2	74.3
JAN [28]	AlexNet	75.2	96.6	99.6	72.8	57.5	56.3	76.3
CDAN [31]	AlexNet	78.3	97.2	100.0	76.3	57.3	57.3	77.7
PFAN [168]	AlexNet	83.0	99.0	99.9	76.3	63.3	60.8	80.4
Source Only	ResNet	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DDC [12]	ResNet	75.6	96.0	98.2	76.5	62.2	61.5	78.3
DAN [13]	ResNet	80.5	97.1	99.6	78.6	63.6	62.8	80.4
RevGrad [163]	ResNet	82.0	96.9	99.1	79.7	68.2	67.4	82.2
JAN [28]	ResNet	86.0	96.7	99.7	85.1	69.2	70.7	84.6
CDAN [31]	ResNet	94.1	98.6	100.0	92.9	71.0	69.3	87.7
TADA [157]	ResNet	94.3	98.7	99.8	91.6	72.9	73.0	88.4
TAT [169]	ResNet	92.5	99.3	100.0	93.2	73.1	72.1	88.4
SHOT [155]	ResNet	90.1	98.4	99.9	94.0	74.7	74.3	88.6
ALDA [170]	ResNet	95.6	97.7	100.0	94.0	72.2	72.5	88.7
Source Only-S	DeiT	86.9	97.7	99.6	87.6	74.9	73.5	86.7
CDTrans-S [41]	DeiT	93.5	98.2	99.6	94.6	78.4	78.0	90.4
Source Only-B	DeiT	90.4	98.2	100.0	90.8	76.8	76.4	88.8
CDTrans-B [41]	DeiT	96.7	99.0	100.0	97.0	81.1	81.9	92.6
Source Only	Swin	89.2	94.1	100.0	93.1	80.9	81.3	89.8
BCAT [42]	Swin	99.2	99.5	100.0	99.6	85.7	86.1	95.0
Source Only	ViT	89.2	98.9	100.0	88.8	80.1	79.8	89.5
Baseline	ViT	91.6	99.0	100.0	90.6	80.2	80.1	90.2
TVT*	ViT	95.7	98.7	100.0	95.4	80.6	80.3	91.8
TVT	ViT	96.4	99.4	100.0	96.4	84.9	86.1	93.9

and testing on the labeled target data. Baseline denotes vanilla ViT with adversarial adaptation [24].

5.5.3 Implementation Details

The ViT-Base with 16×16 input patch size (or ViT-B/16) [40] pre-trained on ImageNet-21K [106] is used as our backbone. The transformer encoder of ViT-B/16 contains 12 transformer layers in total.

Table 5.3: Performance comparison on the Office-Home dataset. TVT* indicates that the backbone is pre-trained on ImageNet. "-S" and "-B" indicate that the backbone is DeiT-Small and DeiT-Base, respectively

Algorithm		A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
Source Only	AlexNet	26.4	32.6	41.3	22.1	41.7	42.1	20.5	20.3	51.1	31.0	27.9	54.9	34.3
DAN [13]	AlexNet	31.7	43.2	55.1	33.8	48.6	50.8	30.1	35.1	57.7	44.6	39.3	63.7	44.5
RevGrad [163]	AlexNet	36.4	45.2	54.7	35.2	51.8	55.1	31.6	39.7	59.3	45.7	46.4	65.9	47.3
JAN [28]	AlexNet	35.5	46.1	57.7	36.4	53.3	54.5	33.4	40.3	60.1	45.9	47.4	67.9	48.2
Source Only	ResNet	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [13]	ResNet	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
RevGrad [163]	ResNet	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [28]	ResNet	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [31]	ResNet	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
TAT [169]	ResNet	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
ALDA [170]	ResNet	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
TADA [157]	ResNet	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SHOT [155]	ResNet	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
Source Only-S	DeiT	55.6	73.0	79.4	70.6	72.9	76.3	67.5	51.0	81.0	74.5	53.2	82.7	69.8
CDTrans-S [41]	DeiT	60.6	79.5	82.4	75.6	81.0	82.3	72.5	56.7	84.4	77.0	59.1	85.5	74.7
WinTR-S [43]	DeiT	65.3	84.1	85.0	76.8	84.5	84.4	73.4	60.0	85.7	77.2	63.1	86.8	77.2
Source Only-B	DeiT	61.8	79.5	84.3	75.4	78.8	81.2	72.8	55.7	84.4	78.3	59.3	86.0	74.8
CDTrans-B [41]	DeiT	68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
Source Only	Swin	64.5	84.8	87.6	82.2	84.6	86.7	78.8	60.3	88.9	82.8	65.3	89.6	79.7
BCAT [42]	Swin	75.3	90.0	92.9	88.6	90.3	92.7	87.4	73.7	92.5	86.7	75.4	93.5	86.6
Source Only	ViT	66.2	84.3	86.6	77.9	83.3	84.3	76.0	62.7	88.7	80.1	66.2	88.7	78.7
Baseline	ViT	71.9	80.7	86.7	79.9	80.4	83.5	76.9	70.9	88.3	83.0	72.9	88.4	80.3
TVT*	ViT	67.1	83.5	87.3	77.4	85.0	85.6	75.6	64.9	86.6	79.1	67.2	88.0	78.9
TVT	ViT	74.9	86.8	89.5	82.8	88.0	88.3	79.8	71.9	90.1	85.5	74.6	90.6	83.6

We train all ViT-based models using mini-batch Stochastic Gradient Descent (SGD) optimizer with the momentum of 0.9. We initialized the learning rate as 0 and linearly increase it to $lr = 0.03$ after 500 training steps. We then decrease it by the cosine decay strategy. The only exception is that we set $lr = 0.003$ for D→ A and W→ A in Office-31 dataset.

Table 5.4: Performance comparison on the VisDA-2017 dataset. TVT* indicates that the backbone is pre-trained on ImageNet. "-B" indicates that the backbone is DeiT-base

Algorithm		plane	bcycl	bus	car	house	knife	mcycl	person	plant	sktbrd	train	truck	Avg
Source Only	ResNet	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
RevGrad [163]	ResNet	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
MCD [165]	ResNet	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
ALDA [170]	ResNet	93.8	74.1	82.4	69.4	90.6	87.2	89.0	67.6	93.4	76.1	87.7	22.2	77.8
DTA [172]	ResNet	93.7	82.2	85.6	83.8	93.0	81.0	90.7	82.1	95.1	78.1	86.4	32.1	81.5
SHOT [155]	ResNet	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
Source Only-B	DeiT	97.7	48.1	86.6	61.6	78.1	63.4	94.7	10.3	87.7	47.7	94.4	35.5	67.1
CDTrans-B [41]	DeiT	97.1	90.5	82.4	77.5	96.6	96.1	93.6	88.6	97.9	86.9	90.3	62.8	88.4
WinTR-B [43]	DeiT	98.7	91.2	93.0	91.9	98.1	96.1	94.0	72.7	97.0	95.5	95.3	57.9	90.1
Source Only	Swin	98.7	63.0	86.7	68.5	94.6	59.4	98.0	22.0	81.9	91.4	96.7	25.7	73.9
BCAT [42]	Swin	99.1	91.6	86.6	72.3	98.7	97.9	96.5	82.3	94.2	96.0	93.9	61.3	89.2
Source Only	ViT	98.2	73.0	82.5	62.0	97.3	63.5	96.5	29.8	68.7	86.7	96.7	23.7	73.2
Baseline	ViT	94.6	81.6	81.8	69.9	93.5	69.9	88.6	50.5	86.8	88.5	91.5	20.1	76.4
TVT*	ViT	97.1	88.8	86.4	64.4	96.4	97.4	90.6	64.1	92.0	90.3	93.7	59.6	85.1
TVT	ViT	97.1	92.9	85.3	66.4	97.1	97.1	89.3	75.5	95.0	94.7	94.5	55.1	86.7

Table 5.5: Ablation study of each module

Methods	Digits	Office-31	Office-Home	VisDA-2017	Avg
Source Only	88.3	89.5	78.7	73.2	82.4
+TAM	97.2	91.2	81.3	79.3	87.3
+DCM	98.9	93.9	83.6	86.7	90.8

5.5.4 Results of Digit Recognition

For the digit recognition task, we perform evaluations on SVHN→MNISt, USPS→MNIST, and MNIST→USPS, following the standard evaluation protocol of UDA. Shown in Table 5.1, TVT obtains the best mean accuracy for each task and outperforms prior work in terms of the average classification accuracy. TVT also performs better than Baseline (+2.7%) due to the contribution of the proposed TAM and DCM. In particular, TVT achieves comparable results to Target Only model, indicating that the domain shift problem is well alleviated.

5.5.5 Results of Object Recognition

For object recognition task, Office-31, Office-Home, and VisDA-2017 are used in evaluation. As shown in Table 5.2 5.3, 5.4, TVT sets up new benchmark results for all the three datasets. On the medium-sized Office-Home dataset (Table 5.3), we achieve the significant improvement over the best prior UDA method (83.6% vs 71.8%).

Results on the large-scale VisDA-2017 dataset (Table 5.4) show that we not only achieve a higher average accuracy, but also compete favorably against ALDA and SHOT. Specifically, we use the most naive pseudo-labeling strategy (pseudo labels with high confidence) [173] in this experiment. Note that DTA also enforces the cluster assumption to learn discriminative features, but it fails to encourage the global diversity which may leads to a degenerate solution where every point is assigned to the same class. Besides, TVT surpasses both Source Only and Baseline, revealing its effectiveness in transferring domain knowledge by (i) capturing both transferable and discriminative fine-grained features and (ii) retaining discriminative information while searching for the domain-invariant representations.

This is also evidenced by the t-SNE visualization of learned features as showcased in Figure 5.2. Obviously, TAM can effectively align source and target domain features by exploiting the local feature transferability. However, the target feature is not well-separated due to that target labels in training are absent and the discriminative information are destroyed by adversarial alignment. Fortunately, this problem is alleviated by DCM by assuming that datapoints should be classified with large margin, as illustrated in Figure 5.2 (D). It is noteworthy that several contemporary work [41, 42, 43] use DeiT [44] or Swin [45] as the backbone and outperforms our method. We argue that this can be mainly explained by the data-efficient merits of DeiT and Swin. Detailed discussion are referred to the supplementary.

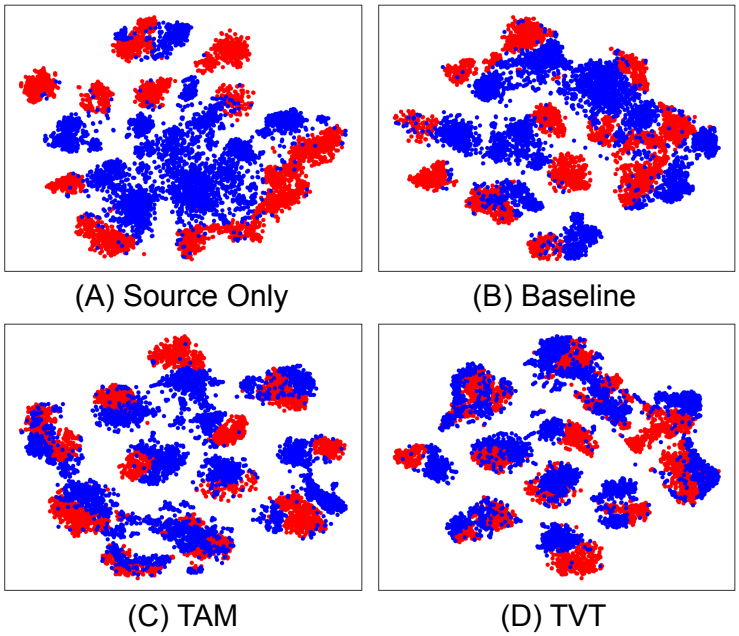

Figure 5.2: t-SNE visualization of VisDA-2017 dataset, where red and blue points indicate the source (synthetic rendering) and the target (real images) domain, respectively

5.5.6 Ablation Study

To learn the individual contribution of TAM and DCM in improving the knowledge transferability of ViT, we conduct the ablation study in Table 5.5. Compared to Source Only, TAM consistently improves the classification accuracy with average 4.9% boost, indicating the significance of capturing both transferable and discriminative features. The performance is further improved by incorporating DCM, justifying the necessary of retaining the discriminative information of the learned representation. It is noteworthy that DCM brings the largest improvement on the large-scale synthetic-to-real VisDA-2017 dataset. We suspect that the large domain gap in VisDA-2017 (synthetic 2D rendering to natural image) is the leading reason, since simply aligning two domains with large domain shift results in a mess distributed feature space.

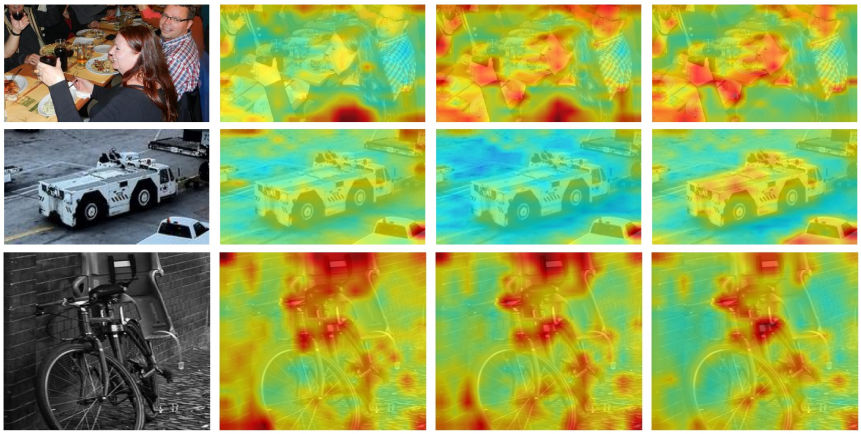

Figure 5.3: Attention map visualization of person, truck, and bicycle in VisDA-2017 dataset. The hotter the color, the higher the attention

This challenge, however, can be largely addressed by DCM that enables retaining discriminative information based on a cluster assumption.

5.5.7 Attention Visualization

We visualize the attention map of the class token in TAM to verify that our model can attend to local features that are both transferable and discriminative. Without loss of generality, we randomly sample target-domain images in VisDA-2017 dataset for comparison. As shown in Figure 5.3, our method captures more accurate regions than Source Only and Baseline. For instance, to recognize the person in the top-left image, Source Only mainly focus on women's shoulder which is discriminative yet not highly transferable. Moving beyond the shoulder region, the baseline also attends to faces and hands that can generalize well across domains. Our method, instead, ignores the shoulder and only highlight those regions that are important for classification and transferable. Certainly, by leveraging the intrinsic attention mechanism and fine-grained features captured by sequential patches, our method promotes the capability of ViT in transferring domain knowledge.

5.6 Summary and Discussion

In this chapter, we perform the first-of-its-kind investigation of ViT's generalization ability in UDA task. To further improve the power of ViT in transferring domain knowledge, we propose TVT by explicitly considering the intrinsic merits of transformer architecture. Specifically, TVT captures both transferable and discriminative features in the given image, and retains discriminative information of the learnt domain-invariant representations. Experimental results on widely used benchmarks show that TVT outperforms prior UDA methods by a large margin.

CHAPTER 6

CONCLUSION AND FUTURE WORK

DNNs have proved their unprecedented power in various applications, such as computer vision, natural language processing, drug discovery, recommendation systems, bioinformatics, and financial fraud detection. However, it is widely recognized that the success of DNNs heavily depends on massive labeled data. This poses a great challenge to some scenarios where the labeled data is not always available. To address this problem, transfer learning, especially UDA, has been proposed and thoroughly investigated in the past decade. Although numerous UDA methods have been proposed and successfully applied to real-world tasks, these methods still suffer from various problems. In this dissertation, we have discussed current research challenges in UDA and proposed new methods to address these challenges.

6.1 Conclusion

6.1.1 Label-Driven Reconstruction for Domain Adaptation in Semantic Segmentation

Although the image-to-image translation strategy reduces the appearance discrepancy between the source domain and the target domain, it also introduces translation bias. Furthermore, existing UDA methods in semantic segmentation fail to ensure prediction consistency in the target domain. Therefore, we propose to use target-to-source translation and reconstruct source and target images from their label space to encourage semantic consistency. More details are referred to Chapter 2.

6.1.2 Context-Aware Domain Adaptation in Semantic Segmentation

It is well known that context dependency is essential for semantic segmentation, while its transferability remains unexplored in UDA. Hereby, we introduce two cross-domain attention modules to capture spatial and channel context between source and target domains. The obtained contextual dependencies can be adapted across domains to facilitate knowledge transfer, which is proved in empirical studies. More details are referred to Chapter 3.

6.1.3 Exploring Robustness of Unsupervised Domain Adaptation in Semantic Segmentation

Despite the success achieved by DNNs, recent studies prove that they are vulnerable to adversarial attacks. Motivated by this observation, we investigate the robustness of existing UDA methods and observe that these models can dramatically degrade under an unnoticeable perturbation. We hereby propose to leverage adversarial examples to improve the robustness of UDA against adversarial attacks. More details are referred to Chapter 4.

6.1.4 TVT: Transferable Vision Transformer for Unsupervised Domain Adaptation

Existing UDA methods are mainly built upon CNNs, while the generalization ability of vision transformer (ViT) is still not well understood. We, therefore, investigate the capability of ViT in transferring knowledge on domain adaptation tasks. Furthermore, we propose a new UDA method that leverages the intrinsic characteristics of ViT, such that our method can capture both transferable and discriminative features for domain adaptation. More details are referred to Chapter 5.

6.2 Future Work

In terms of future works, a few open challenges and directions are outlined as follows. One such direction is to design more advanced pseudo labeling strategies to improve the accuracy of the target pseudo labels. The rationale is that pseudo labeling plays an important role in recent UDA methods. Therefore, more accurate target pseudo labels are expected to improve the performance further. Although previous studies apply various strategies to address this issue, they still suffer from severe inaccuracy problems, which are problematic.

Another direction is to explore the attention mechanisms of transformer architectures. It is well-known that cross-attention is good at aligning different distributions, such as distributions from different domains. As a preliminary study, our recent work [20] demonstrates the power of cross-attention in UDA. However, the combination of cross-attention and UDA is largely unexplored, suggesting great potential in the future.

More generally, we should consider domains from different modalities, such as vision-language adaptation. Our preliminary studies suggest that visual features and linguistic features can be aligned well in the embedding space [174, 175]. Research along this direction would be incredibly valuable and more generalizable.

REFERENCES

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

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

[3] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[4] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.

[5] Y. Park and M. Kellis, "Deep learning for regulatory genomics," *Nature biotechnology*, vol. 33, no. 8, pp. 825–826, 2015.

[6] J. Yang, A. Ma, A. D. Hoppe, C. Wang, Y. Li, C. Zhang, Y. Wang, B. Liu, and Q. Ma, "Prediction of regulatory motifs from human chip-sequencing data using a deep learning framework," *Nucleic acids research*, vol. 47, no. 15, pp. 7809–7824, 2019.

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

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[9] Q. Dou, C. Ouyang, C. Chen, H. Chen, and P.-A. Heng, "Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss," *arXiv preprint arXiv:1804.10916*, 2018.

[10] C. Chen, Q. Dou, H. Chen, J. Qin, and P.-A. Heng, "Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 865–872.

[11] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," *arXiv preprint arXiv:2102.09508*, 2021.

[12] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[13] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning (ICML)*, 2015.

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

[15] Y. Zhang, Z. Wang, and Y. Mao, "Rpn prototype alignment for domain adaptive object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 425–12 434.

[16] V. VS, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, "Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4516–4526.

[17] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[18] "Cycada: Cycle consistent adversarial domain adaptation," in *International Conference on Machine Learning (ICML)*, 2018.

[19] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] J. Yang, W. An, C. Yan, P. Zhao, and J. Huang, "Context-aware domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[21] J. Yang, C. Li, W. An, H. Ma, Y. Guo, Y. Rong, P. Zhao, and J. Huang, "Exploring robustness of unsupervised domain adaptation in semantic segmentation," *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2021.

[22] J. Yang, J. Liu, N. Xu, and J. Huang, "Tvt: Transferable vision transformer for unsupervised domain adaptation," *arXiv preprint arXiv:2108.05988*, 2021.

[23] A. Ramponi and B. Plank, "Neural unsupervised domain adaptation in nlp—a survey," *arXiv preprint arXiv:2006.00632*, 2020.

[24] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[25] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International Conference on Machine Learning (ICML)*, 2017.

[29] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

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

[31] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *arXiv preprint arXiv:1705.10667*, 2017.

[32] X. Zhu, H. Zhou, C. Yang, J. Shi, and D. Lin, "Penalizing top performers: Conservative loss for semantic segmentation adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 568–583.

[33] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 13, 2018.

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

[35] J. Huang, S. Lu, D. Guan, and X. Zhang, "Contextual-relation consistent domain adaptation for semantic segmentation," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[36] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3764–3773.

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

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

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

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[41] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, and R. Jin, "Cdtrans: Cross-domain transformer for unsupervised domain adaptation," *arXiv preprint arXiv:2109.06165*, 2021.

[42] X. Wang, P. Guo, and Y. Zhang, "Domain adaptation via bidirectional cross-attention transformer," *arXiv preprint arXiv:2201.05887*, 2022.

[43] W. Ma, J. Zhang, S. Li, C. H. Liu, Y. Wang, and W. Li, "Exploiting both domain-specific and invariant knowledge via a win-win transformer for unsupervised domain adaptation," *arXiv preprint arXiv:2111.12941*, 2021.

[44] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[46] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang, "Label-driven reconstruction for domain adaptation in semantic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 480–498.

[47] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015, pp. 1440–1448.

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

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

[50] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.

[51] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017.

[52] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019.

[53] Y. Chen, W. Li, X. Chen, and L. V. Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1841–1850.

[54] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Dada: Depth-aware domain adaptation in semantic segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019.

[55] Y. Zou, Z. Yu, B. V. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 297–313.

[56] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, October 2019.

[57] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2507–2516.

[58] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019.

[59] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6810–6818.

[60] Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 518–534.

[61] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4500–4509.

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

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

[64] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in neural information processing systems (NIPS)*, 2017, pp. 700–708.

[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[66] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[67] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *International Conference on Learning Representations (ICLR)*, 2014.

[68] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015, pp. 1377–1385.

[69] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[70] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, vol. 40, no. 4, pp. 834–848, 2018.

[71] L.-C. Chen, M. D. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens, "Searching for efficient multi-scale architectures for dense image prediction," in *Advances in neural information processing systems (NIPS)*, 2018.

[72] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015, pp. 1635–1643.

[73] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1713–1721.

[74] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning (ICML)*, 2015.

[75] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015, pp. 4068–4076.

[76] W. Ying, Y. Zhang, J. Huang, and Q. Yang, "Transfer learning via learning to transfer," in *International Conference on Machine Learning (ICML)*, 2018, pp. 5072–5081.

[77] Y. Zhang, Y. Wei, Q. Wu, P. Zhao, S. Niu, J. Huang, and M. Tan, "Collaborative unsupervised domain adaptation for medical image diagnosis," *IEEE Transaction on Image Processing (TIP)*, 2020.

[78] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 597–613.

[79] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Advances in neural information processing systems (NIPS)*, 2016, pp. 343–351.

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

[81] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[82] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[83] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.

[84] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 694–711.

[85] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[86] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[87] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[88] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2014.

[89] Y. Chen, W. Li, and L. Van Gool, "Road: Reality oriented adaptation for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7892–7901.

[90] K.-H. Lee, G. Ros, J. Li, and A. Gaidon, "Spigan: Privileged adversarial learning from simulation," in *International Conference on Learning Representations (ICLR)*, 2019.

[91] J. Choi, T. Kim, and C. Kim, "Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019.

[92] Y. Jang, H. Lee, S. J. Hwang, and J. Shin, "Learning what and where to transfer," in *International Conference on Machine Learning (ICML)*, 2019.

[93] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[94] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[95] R. Sun, X. Zhu, C. Wu, C. Huang, J. Shi, and L. Ma, "Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection," in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[96] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7151–7160.

[97] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 267–283.

[98] H. Wang, T. Shen, W. Zhang, L. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[99] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 891–898.

[100] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[101] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems (NIPS)*, 2017, pp. 5998–6008.

[102] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *CoRR*, vol. abs/1805.08318, 2018.

[103] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *International Conference on Learning Representations (ICLR)*, 2017.

[104] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3146–3154.

[105] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

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

[107] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning (ICML)*, 2013.

[108] H. X. Minghao Chen and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2019.

[109] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting deeper into the future of semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[110] A. Raju, C.-T. Cheng, Y. Huo, J. Cai, J. Huang, J. Xiao, L. Lu, C. Liao, and A. P. Harrison, "Co-heterogeneous and adaptive segmentation from multi-source and multi-phase ct imaging data: A study on pathological liver and lesion segmentation," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[111] I. Shin, S. Woo, F. Pan, and I. S. Kweon, "Two-phase pseudo label densification for self-training based domain adaptation," *Proceedings of the European Conference on Computer Vision*, 2020.

[112] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[113] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "Darts: Deceiving autonomous cars with toxic signs," *arXiv preprint arXiv:1802.06430*, 2018.

[114] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2015.

[115] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *International Conference on Learning Representations (ICLR)*, 2018.

[116] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[117] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *International Conference on Machine Learning (ICML)*, 2020.

[118] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[119] X. Zhan, Z. Liu, P. Luo, X. Tang, and C. C. Loy, "Mix-and-match tuning for self-supervised semantic segmentation," *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[120] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2015.

[121] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision (ECCV)*, 2016.

[122] J. Xu, L. Xiao, and A. M. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, 2019.

[123] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[124] C.-H. Ho and N. Vasconcelos, "Contrastive learning with adversarial examples," *Advances in neural information processing systems (NeurIPS)*, 2020.

[125] Z. Jiang, T. Chen, T. Chen, and Z. Wang, "Robust pre-training by adversarial contrastive learning." in *NeurIPS*, 2020.

[126] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, and L. Lin, "An adversarial perturbation oriented domain adaptation approach for semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[127] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *International conference on machine learning (ICML)*, 2012.

[128] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[129] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *International Conference on Learning Representations (ICLR)*, 2014.

[130] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Conference on Learning Representations (ICLR)*, 2015.

[131] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[132] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[133] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[134] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Learning Representations (ICLR)*, 2018.

[135] Y. Yang, D. Lao, G. Sundaramoorthi, and S. Soatto, "Phase consistent ecological domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[136] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *International Conference on Learning Representations (ICLR)*, 2019.

[137] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*. PMLR, 2019.

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

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

[140] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *arXiv preprint arXiv:2102.12122*, 2021.

[141] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *arXiv preprint arXiv:2012.15840*, 2020.

[142] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 244–253.

[143] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," *arXiv preprint arXiv:2102.00719*, 2021.

[144] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[145] H.-Y. Zhou, C. Lu, S. Yang, and Y. Yu, "Convnets vs. transformers: Whose visual representations are more transferable?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2230–2238.

[146] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *arXiv preprint arXiv:1411.1792*, 2014.

[147] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[148] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[149] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," *arXiv preprint arXiv:2012.07436*, 2020.

[150] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[151] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3286–3295.

[152] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3588–3597.

[153] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[154] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," *arXiv preprint arXiv:2011.14503*, 2020.

[155] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6028–6039.

[156] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[157] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[158] J. S. Bridle, A. J. Heading, and D. J. MacKay, "Unsupervised classifiers, mutual information and'phantom targets'," 1992.

[159] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *International workshop on artificial intelligence and statistics*. PMLR, 2005, pp. 57–64.

[160] R. Gomes, A. Krause, and P. Perona, "Discriminative clustering by regularized information maximization," 2010.

[161] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," *arXiv preprint arXiv:1206.6438*, 2012.

[162] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," in *International conference on machine learning*. PMLR, 2017, pp. 1558–1567.

[163] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[164] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*. PMLR, 2018, pp. 1989–1998.

[165] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3723–3732.

[166] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.

[167] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[168] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 627–636.

[169] H. Liu, M. Long, J. Wang, and M. Jordan, "Transferable adversarial training: A general approach to adapting deep classifiers," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4013–4022.

[170] M. Chen, S. Zhao, H. Liu, and D. Cai, "Adversarial-learned loss for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[171] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," *arXiv preprint arXiv:1710.06924*, 2017.

[172] S. Lee, D. Kim, N. Kim, and S.-G. Jeong, "Drop to adapt: Learning discriminative features for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 91–100.

[173] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[174] J. Yang, J. Duan, S. Tran, Y. Xu, S. Chanda, L. Chen, B. Zeng, T. Chilimbi, and J. Huang, "Vision-language pre-training with triple contrastive learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[175] J. Duan, L. Chen, S. Tran, J. Yang, Y. Xu, B. Zeng, C. Tao, and T. Chilimbi, "Multi-modal alignment using representation codebook," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

BIOGRAPHICAL STATEMENT

Jinyu Yang received his Ph.D. degree in Computer Science from the University of Texas at Arlington in 2022. Prior to the Ph.D. program at UTA, he received his M.S degree in Statistics from South Dakota State University in 2017. He received his B.S degree in computer science from Jilin University in 2015. His research mainly lies in the areas of machine learning and computer vision, with a particular focus on transfer learning, self-supervised learning, vision-language pretraining, graph representation learning, and adversarial learning. He is interested in developing algorithms that equip machines with more general intelligence via knowledge transfer or self-supervision, such that the learned knowledge can be generalized well to different domains or various downstream tasks. During his Ph.D. study, he has published more than 10 top-tier conference and journal papers, such as IEEE Conference on Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), AAAI Conference on Artificial Intelligence (AAAI), Annual Conference on Neural Information Processing Systems (NeurIPS), IEEE Winter Conference on Applications of Computer Vision (WACV), Nucleic Acids Research, and Bioinformatics. He has been invited as a reviewer for many top-tier conferences and journals, such as Computational Biology and Chemistry, IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), ICIBM, NeurIPS, ICLR, WACV, CVPR, ICML, ECCV, and MICCAI.