THE UNIVERSITY of ADELAIDE

Relaxed Invariant Representation for Unsupervised Domain Adaptation

by
Hossein Askari Lyarjdameh

A thesis submitted in fulfillment for the degree of
Master of Philosophy

in

School of Computer Science
Faculty of Engineering, Computer & Mathematical Sciences
The University of Adelaide

October 5, 2021

Contents

List of Figures

List of Tables

Abstract

The success of supervised machine learning relies on the availability of a large amount of annotated training data from different domains, which is often cost-ineffective to collect, and unrealistic in many scenarios. Unsupervised domain adaptation (UDA) aims to overcome this problem by transferring predictive models trained on a labelled source domain to an unlabelled target domain, with the difficulty of resolving distributional shift between domains. To bridge this distribution gap, recent advances in deep learning focus on learning representations that are invariant across domains. However, such an approach may fail to generalize well to target domains and may even considerably deteriorate adaptability, due to the existence of an inherent trade-off between adaptability and invariance. Building on advances in deep generative models, this thesis aims to relax the learning of invariant representations, and to develop efficient algorithms for UDA.

This thesis comprises two parts. The first part introduces the problem of learning invariant representations. In particular, we mathematically derive a lower bound on the joint probability distribution of the source and target domains as a framework for UDA and theoretically discuss how this bound can be used to relax the invariance in representation learning. Following this motivation, in the second part, we design a simple, yet efficient algorithm to address the challenges of forcing too much invariance in domain distributional matching. We empirically show how the trade-off between adaptability and invariant representation can be mitigated with an invertible architecture between the representation and predictor models while learning the invariant representation. The experiments are run on public benchmark problems and the results show that the proposed method relaxes the excessive invariance effectively and outperforms the existing domain adaptation approaches.

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Adelaide International Scholarship (AIS).

Hossein Askari Lyarjdameh

16-July-2021

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Professor Gustavo Carneiro and Professor Ian Reid, for their tremendous dedication, unwavering support, and an infinite source of inspiration.

I wish to extend my deepest and enduring appreciation to my principal supervisor, Professor Gustavo Carneiro, for providing me with invaluable guidance, insightful suggestions, and constructive criticisms throughout this research. He motivated me to grow into an independent researcher and supported me all along the way. It was a great privilege and honor to do research under his supervision.

I would like to thank Professor Ian Reid for his constructive comments and suggestions on my work. I would also like to thank Dr. Ehsan Abbasnejad for his assistance at the early stage of my research.

I gratefully acknowledge the University of Adelaide for awarding me a scholarship to pursue my studies towards a higher degree by research. I also acknowledge the school of Computer Science and Australian Institute for Machine Learning (AIML), for providing students with a friendly atmosphere to do research. Special thanks to the all staff and members of AIML for their support and kindness.

Last but not least, my heartfelt gratitude goes to my mother, my late father, and my siblings for their love, dedication, and unconditional support throughout my life.

Chapter 1

Introduction

1.1 Overview and Research Problem

In recent years, machine learning (ML) has produced powerful methods that enable computers to observe the world, learn from that observation, and ultimately perform automated decision making. Deep learning (DL) models are currently the most widespread and successful methodology in ML. In fact, deep learning is an advanced method in ML for computational learning of high-level concepts using deep hierarchical neural networks (LeCun, Bengio, and Hinton, 2015). DL models have demonstrated outstanding performance on challenging ML tasks, such as the ImageNet recognition task (Krizhevsky, Sutskever, and Hinton, 2012), the board game Go (Silver et al., 2017), the Stanford question answering competition (Rajpurkar et al., 2016), medical imaging (Zeng et al., 2019), biological sequence analysis (Iuchi et al., 2021), and self-driving cars (Yue et al., 2018). They have also caught the attention of many industry communities. For instance, Google leverages deep learning for image search (Szegedy et al., 2015), Facebook utilizes deep learning for automatic tagging, and Amazon develops deep learning methods for product recommendations (Goodfellow et al., 2016).

Despite these tremendous advances, training deep models heavily rely on the availability of large-scale, labelled dataset. In many cases, obtaining a sufficient amount of annotated data tends to require a massive amount of computational and human resources, constraining the type of problems that can be addressed. Hence, leveraging another, but related labelled training dataset becomes a promising solution. With the explosive growth of various sources of data on the Internet, an incredibly large number of labelled datasets can be readily exploited. In this case, it makes sense to train a model with multi-source web data and translate it to the target data. A prevailing issue, however, is the distribution mismatch between the source and target domains, which can

severely undermine the model performance. This is due to the fact that the source and target data are not independent and identically distributed (i.i.d), thus violating the assumption that these datasets come from the same distribution, and models fail to generalize well to new testing domains. As an example, in medical imaging, clinicians manually annotate tissues and abnormalities to form training data for computer-aided-diagnosis (CAD) systems. But, owing mostly to the calibration, the mechanical and electrical configuration, and acquisition protocol of scanners, there are wide variations between data sets from different medical centres. Consequently, CAD systems trained by a dataset acquired from centre A, typically performs poorly on test datasets obtained from centre B. Even slight deviations between training distributions can give rise to significant performance deterioration. This problem is crucial in computer vision applications, as datasets can be significantly different because of a variety of factors, such as camera pose, object scale, illumination, camera characteristics, labeling process, image selection process, and so on. This inability to generalize out of the training distribution hinders the safe deployment of DL models in real-world, high-stakes settings such as medical diagnosis, criminal justice, and autonomous vehicles. Transfer learning (TL) has been introduced to cope with the generalization problem stemming from distributional shift. TL works by finding a transformation from the labelled training data, referred to as source domain, to the test data of interest, called target domain.

Mathematically, this distributional shift between source domain and target domain is characterized by the difference in joint probability distribution $p_t(x, y) \neq p_s(x, y)$, where x as input image and y as label represent samples of two random variables $\mathbf{x}$, $\mathbf{y}$ from spaces $\mathcal{X}$, $\mathcal{Y}$. Depending on the availability of labels, the consistency between the feature spaces of the two domains, and the availability of target domain during training, several scenarios are defined and conceptualized in TL. One of these well-studied scenarios is domain adaptation (DA), where the assumption is that the feature spaces of two domains are the same, and the target domain samples, whether unlabelled or partially labelled, and labelled source domain samples are available during training. The objective of DA in this thesis is the learning of a classifier using unlabelled samples from the target domain by leveraging labelled source domain samples–this problem is called unsupervised domain adaptation (UDA).

To tackle the UDA problem, classes of algorithms under different assumptions, and based on well-studied theoretical bounds, have been proposed (Ben-David

et al., 2007; Ben-David et al., 2010; Le et al., 2018; Zhang et al., 2019c; Zhao et al., 2019). Most modern DL approaches to UDA are based on an optimization that minimizes an empirical estimate of some distance between two domains, which leads to learning an intermediate representation that is invariant to the changes between source and target domains (i.e. invariant representation). In other words, the classifier trained on representations of source labelled data will work similarly for the unlabelled target data. This approach may increase the transferability of features, as high transferability is close to an invariant representation whereas low transferability implies features that are more domain specific. Nevertheless, transferability comes at a cost, i.e. it can hurt the discriminability of the representation (adaptability). Note that, discriminability refers to the ability of the model to capture information that is relevant to discriminate the classes. To be more specific, there is a fundamental trade-off between invariance of learned representations and their adaptability in the presence of label shift. Thus, invariant representation learning most likely fails to guarantee a good generalization for the target domain (low target risk). By way of illustration, consider the setting of binary classification with non-overlapping support. Suppose we have two classes of human/vehicle for the source and target domains. As illustrated in Figure 1.1, in input space, it is easy to find a classifier that achieves 100% accuracy on each domain, separately. However, under a translation between two distributions that achieves a perfect alignment (i.e. an invariant transformation), the classifier trained in one domain works poorly on the other domain. In other words, the lower the source error, the higher the target error.

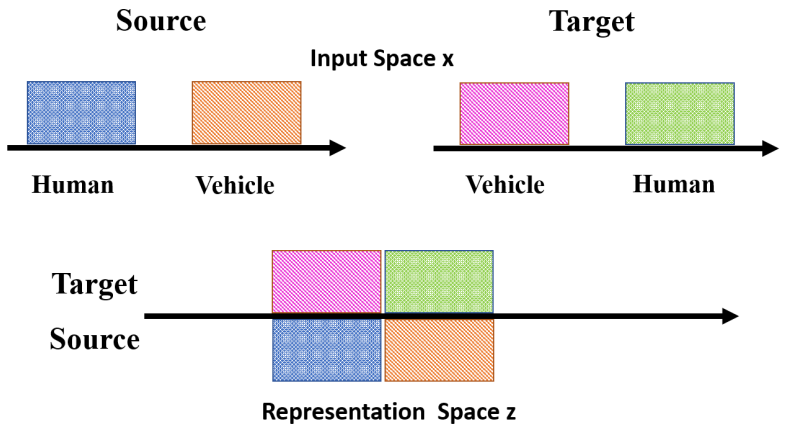


FIGURE 1.1. Failure case of invariant representation learning. The Colored rectangles denote categories, fill patterns denote class-types.

While most previous works deal with this trade-off through i.i.d. assumptions and data augmentation, in this thesis, we address that by relaxing domain invariance.

1.2 Aims and Objectives

The overall aim of this thesis is to address a crucial problem in the field of machine learning and computer vision: domain adaptation, which in this context means the robustness to the data shift between the source and target domains. We will exploit flow-based deep generative models to address the generalization under the unsupervised domain adaptation paradigm (UDA). To achieve this goal, we set the following research objectives.

1. Review the advances of deep UDA methodologies, and discuss the essential problems and the technical challenges of UDA. We identify and critically review a broad solution of UDA proposed by researchers. This review helps us to comprehensively understand the research problem, research status, and theoretical arguments.

2. Demonstrate the need for relaxing the invariance assumption, and formalize a lower bound on the joint probability distribution of source and target domains as a unified framework for UDA. We theoretically discuss how this bound helps to relax the invariance in representation learning. Although not empirically evaluated, this lower bound provides us with a new insight into UDA.

3. Develop an efficient algorithm for UDA through a relaxed version of distribution matching that addresses the trade-off between adaptability and invariance of learned representation. The proposed model is expected to enhance the adaptability (expressed as minimal joint error), by relaxing the domain-invariance. Our method is flexible and easy to implement, and can be deployed as a component of existing DL models to improve both transferability and discriminability of representations. We empirically demonstrate the benefit of our proposed model on public benchmark datasets.

1.3 Thesis Outline

The rest of the thesis is organised as follows:

In Chapter 2, we provide an overview of UDA covering a preliminary concept of UDA (formal definition, notation, and assumptions), which is utilized for unsupervised representation learning, and for theoretical bounds. We also review related, recent papers on single-source, single-target UDA, in which we have one labelled source domain and one unlabelled target domain.

In Chapter 3, the necessity for relaxing invariance on learned representation is discussed and a general relaxed-invariance framework for UDA is introduced accordingly. To this end, the variational inference learning is employed to effectively approximate the intractable joint probability distribution. The intractability stems from the inaccessibility of the target labels.

In Chapter 4, we present a relaxed-invariance version of standard domain adversarial learning for domain adaptation, which incorporates both weighted and invariant representation. The proposed method relies on the meaningful and inductive design of weights on representation invariance. An empirical illustration of our method is provided on benchmark datasets, which validates its performance, effectiveness and versatility.

Chapter 5 concludes the thesis by summarizing the contributions and presenting future work.

Chapter 2

Background and Literature Review

In this chapter, we aim to provide a general review of unsupervised domain adaptation. To this end, we first provide a background on unsupervised representation learning in Sec. 2.1.1, basic concepts of transfer learning in Sec. 2.1.2, and formal definition and categorization of domain adaptation, which can be recognized as a special type of transfer learning, in Sec. 2.1.3. We then define the UDA problem in Sec. 2.2.1, review prominent theoretical bounds in Sec 2.2.2, and review UDA methods proposed by researchers in Sec. 2.2.3.

2.1 Background

2.1.1 Unsupervised Representation Learning

Representation learning from unlabelled data is a well studied problem in machine learning. The classical methods to unsupervised representation learning are based on clustering on the data manifold (for example using c-means)– these clusters are then used to improve classification accuracy (Radford, Metz, and Chintala, 2015). In the context of computer vision, hierarchical clustering (Coates and Ng, 2012) and auto-encoders (Vincent et al., 2010) have been used to learn powerful representations for high-dimensional images. Another way that enables us to learn powerful image representation in an unsupervised manner is by using a deep generative model (DGM). DGMs are designed to encode the underlying probability distributions over data manifolds. The development of DGM has enabled unprecedented performance in a vast range of machine learning tasks such as density estimation, unsupervised representation learning, distribution alignment, image-to-image translation, synthesis, and many other tasks. Mathematically, DGMs aim at estimating an approximation to the generative model by minimizing the distance between the generative distribution

and the data distribution under a certain measure or divergence D,

$$\min_{\theta} \; D(p_{data}(\mathbf{x})||p_{model}(\mathbf{x};\theta)), \tag{2.1}$$

where p_{data} is approximated with empirical data distribution $p_{data}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}(x)$ based on observations $\{x_i\}_{i=1}^{N}$, with $\delta_{x_i}(x)$ defined as:

$$\delta_{x_i}(x) = \begin{cases} 1 & \text{if} \quad x = x_i \\ 0 & \quad \text{otherwise} \end{cases} \tag{2.2}$$

One of the most common statistical divergence measure for probability distribution is Kullback-Leibler (KL) divergence. In practice, the minimization of KL divergence approximates the maximization of the log-likelihood:

$$\begin{aligned} \min_{\theta} \; KL(p_{data}(\mathbf{x})||p_{model}(\mathbf{x};\theta)) &= \min_{\theta} \; \mathbb{E}_{\mathbf{x}\sim p_{data}}\left[\log \frac{p_{data}(\mathbf{x})}{p_{model}(\mathbf{x};\theta)}\right], \\ &\approx \max_{\theta} \; \mathbb{E}_{\mathbf{x}\sim p_{data}} \log p_{model}(\mathbf{x};\theta). \end{aligned} \tag{2.3}$$

Numerous types of algorithms have been proposed in the literature to learn the model $p_{model}(\mathbf{x};\theta)$. In this section, we discuss the following deep generative models: (i) Variational Autoencoders (VAEs) (Kingma and Welling, 2013); (ii) Generative Adversarial Networks (GANs) (Goodfellow et al., 2014); and (iii) Normalizing Flows (NFs) (Dinh, Krueger, and Bengio, 2014; Rezende and Mohamed, 2015).

2.1.1.1 Variational Autoencoder (VAE)

Latent variables refer to those variables that are included in the model, but not observable. In the case of unconditional modelling of observed variable $\mathbf{x}$, the Bayesian graphical model then represents a joint distribution over both the observed variable $\mathbf{x}$ and the latent variable $\mathbf{z}$. The marginal probability distribution over the observed variables $p_\theta(\mathbf{x})$ can be defined as:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x},\mathbf{z})\, d\mathbf{z}. \tag{2.4}$$

The most common latent variable model is specified as factorization with the following probabilistic structure:

$$p_\theta(\mathbf{x},\mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}), \tag{2.5}$$

where $p(\mathbf{z})$ is called prior distribution. The goal of generative modeling is to maximize the log-likelihood of the marginal probability of data,

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})\, d\mathbf{z}, \tag{2.6}$$

but this integral is intractable, which leads to the intractability of the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ because of the following identity

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x})}. \tag{2.7}$$

Thus, a tractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$ needs a tractable marginal likelihood $p_\theta(\mathbf{x})$ and vice versa. Approximate inference uses another distribution $q_\phi(\mathbf{z}|\mathbf{x})$ that makes the computation of the integral in Equation 2.6 tractable and has small approximation error to the exact integral. For any choice of inference model $q_\phi(\mathbf{z}|\mathbf{x})$ with parameters ϕ, the following equation can be derived:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x})\big] && (2.8)\\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\right] && (2.9)\\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\right] && (2.10)\\
&= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\theta,\phi}(\mathbf{x})} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right]\right]}_{KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))}, && (2.11)
\end{aligned}$$

where the second term in Equation 2.11 is the KL divergence between true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ and variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$, which is non-negative. From Equation 2.11 we have:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \log p_\theta(\mathbf{x}) - KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) \leq \log p_\theta(\mathbf{x}). \tag{2.12}$$

The first term in Equation 2.11, labelled as $\mathcal{L}_{\theta,\phi}$, is the variational lower bound, also called evidence lower bound (ELBO).

There are various ways of optimizing the ELBO loss, but for continuous $\mathbf{z}$ this could be done efficiently through the *reparametrization* of variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$, which leads to the variational autoencoder method (Kingma and Welling, 2013).

2.1.1.2 Generative Adversarial Network (GAN)

Another recently developed framework for learning a generative model is based on the Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). Specifically, GAN involves a generator network G, and a discriminator network D, where the generator is trained to map random noise from an arbitrary latent distribution to the data samples, while the discriminator's purpose is to distinguish between real and generated (fake) samples. Indeed, the generator's purpose is to "fool" the discriminator by producing samples that are as similar to the real data as possible. Mathematically, the GAN objective seeks to find a Nash equilibrium to a two-player (G-D) min-max problem,

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\Big[\log D(\mathbf{x})\Big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\Big[\log\left(1 - D(G(\mathbf{z}))\right)\Big], \tag{2.13}$$

where $p(\mathbf{x})$ denotes the true distribution of data, and $\mathbf{z} \in \mathcal{R}^{d_\mathbf{z}}$ is a latent variable sampled from the known distribution $p(\mathbf{z})$ such as $\mathcal{U}[-1, 1]$ or $\mathcal{N}(0, I)$. The generator G implicitly learns how to generate samples from $p(\mathbf{x})$ using samples $\mathbf{z} \sim p(\mathbf{z})$. If the generator G is fixed, the optimal discriminator is $D^*(\mathbf{x}) = \frac{p(\mathbf{x})}{p_g(\mathbf{x})+p(\mathbf{x})}$. When the assumption of an optimal discriminator is true, the generator minimizes the Jensen-Shannon divergence between $p_g(\mathbf{x})$ and $p(\mathbf{x})$. The global equilibrium can be met if $p(\mathbf{x}) = p_g(\mathbf{x})$, and the optimal value of Equation 2.13 is -2 log 2.

The recent research has concentrated on modifications to the GAN procedure by improving the generator (Radford, Metz, and Chintala, 2015), discriminator (Zhao, Mathieu, and LeCun, 2016; Peng et al., 2018), objective loss (Arjovsky, 2017; Lim and Ye, 2017), or the training stability (Salimans et al., 2016; Adler and Lunz, 2018). More recently, within the adversarial learning paradigm, researchers (Dumoulin et al., 2016; Donahue, Krähenbühl, and Darrell, 2016) used a bidirectional network structure, which tries to match the joint distributions of two domains. However, the non-identifiability issues for joint distribution matching are raised by Li et al. (2017). These problems are alleviated in DiscoGAN (Kim et al., 2017), and Cycle-GAN (Zhu et al., 2017) via additional l_1, l_2, or adversarial losses.

2.1.1.3 Normalizing Flow (NF)

The normalizing flow (Dinh, Sohl-Dickstein, and Bengio, 2016) is a likelihood-based generative model defined as an invertible mapping, $f : \mathcal{X} \to \mathcal{Z}$ from

the observed space $\mathcal{X}$ to the latent space $\mathcal{Z}$. The distribution of the observed variable can be modeled by applying a chain of invertible transformations, which is composed of a sequence of invertible functions $g = g_1 \circ g_2 \circ ... \circ g_L : \mathbb{R}^d \to \mathbb{R}^d$ with inverse $f = g^{-1}$, on random latent variables with known distribution $\mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z})$. Based on the change of variables formula, the probability distribution of transformed random variable can be written as follows:

$$p_{\mathcal{X}}(\mathbf{x}) = p_{\mathcal{Z}}(f(\mathbf{x})) \left| \det(J_f(\mathbf{x})) \right| = p_{\mathcal{Z}}(f(\mathbf{x})) \prod_{l=1}^{L} \left| \det(J_{f_l}(\mathbf{h}_l)) \right|, \tag{2.14}$$

where $J_f(\mathbf{x}) = \partial f(\mathbf{x})/\partial \mathbf{x}$ is the Jacobian of f with respect to $\mathbf{x}$, $\det(.)$ denotes the determinant, and $\mathbf{h}_l$ denotes the output of intermediate mapping g_l, with $\mathbf{h}_1 = \mathbf{x}$ and $\mathbf{h}_L = g_L(\mathbf{z})$. The mapping $f(\mathbf{x})$ is characterized by a neural network with an architecture that is designed to ensure the invertibility and efficient computation of determinants. We train the model by computing the negative log-likelihood of the training data $D = \{\mathbf{x}_i\}_{i=1}^N$ with respect to the parameters η.

$$\eta^* = \underset{\eta}{\operatorname{argmax}}\, \mathcal{L}, \; \mathcal{L} = -\frac{1}{|D|} \sum_{\mathbf{x} \in D} \log p(\mathbf{x}; \eta) \tag{2.15}$$

2.1.2 Transfer Learning (TL)

Two paramount problems facing machine learning are concerned with data-efficiency and generalization. The former one assumes the model should be able to learn from only a few datapoints, while the latter refers to the robustness to data shifts (distributional shift) (Kingma and Dhariwal, 2018). As for generalization, intelligent systems equipped with machine learning algorithms often perform poorly when training and testing datapoints are drawn from different probability distributions (Zhao et al., 2019). By way of illustration, a prognosis system trained by the labelled data collected from an Australian hospital, may not work well on the data samples acquired from a European hospital. The system, therefore, is required to be either trained from scratch, or fine-tuned using labelled data from the new domain, which is costly to acquire. This generalizability problem has given rise to a new research problem in machine learning characterized as transfer learning. Inspired from human being's adaptability to new domains, transfer learning aims at generalizing by transferring knowledge across distributions. This is accomplished with the use of knowledge from domains with abundant labels to train a predictor for the domain with insufficient labels.

In what follows, we introduce the essential preliminary definitions that formalize the concept of transfer learning.

Definition 2.1. **(*Domain*)** Domain corresponds to the marginal distribution $\mathcal{D}$ on the input space $\mathcal{X} \subset \mathbb{R}^d$, and a labelling function $f : \mathcal{X} \to \mathcal{C}$ that maps the input space to the classes $\mathcal{C} = \{1, ..., C\}$.

Definition 2.2. **(*Transfer Learning*)** Let us consider $\langle \mathcal{D}_s, f_s \rangle$ and $\langle \mathcal{D}_t, f_t \rangle$ to be the source and target domains, respectively. Transfer learning aims at improving the target labelling function f_t by using $\mathcal{D}_s$, $\mathcal{D}_t$, and f_s.

According to the aforementioned definitions, several possible learning settings based on the label-setting, consistency between the feature spaces of two domains, and the availability of target domain during training can be defined in transfer learning. A complete list of these learning settings is categorized and presented by Zhang et al. (2019a).

2.1.3 Domain Adaptation (DA)

A particularly interesting subfield of transfer learning is domain adaptation, where we assume that the feature space in both source and target domains and the label space in their corresponding learning tasks remain stationary. The objective in domain adaptation is to learn a predictor in the presence of a shift between the source (training) and target (test) data distributions. Depending on the number of available labelled samples in the target domain, the label space differences across domains, and the number of source and target domains, various scenarios are considered in the literature of domain adaptation (You et al., 2019; Cao et al., 2018; Motiian et al., 2017; Saito et al., 2019).

Let the number of samples in the target domain be N_t and the number of those samples that are labelled be N_{tl}; then, DA can be categorized into: (i) semi-supervised DA, when $N_{tl} < N_t$, (ii) unsupervised DA when $N_{tl} = 0$, (iii) few-shot DA when we have $N_{tl} < N_t$, and $N_{tl} < 20$.

Let $\mathcal{C}_s$ and $\mathcal{C}_t$ be the the set of labels for the source and target domains respectively; then, domain adaptation can be categorized into: (i) closed-set DA, when $\mathcal{C}_s = \mathcal{C}_t$; (ii) open-set DA, when $\mathcal{C}_s \subset \mathcal{C}_t$, that is, when source label set is a proper subset of target label set; (iii) partial DA, when $\mathcal{C}_t \subset \mathcal{C}_s$, that is, when target label set is a subset of source label set; and (iv) universal DA, where the

prior knowledge of the label sets is unavailable.

Furthermore, suppose the number of source domains and target domains is K_s, and K_t respectively. Then, the DA tasks can be categorized into: (i) single-source DA, when $K_s = 1$; (ii) multi-source DA, if $K_s > 1$. (iii) single-target DA, when $K_t = 1$; (iv) multi-target DA, if $K_t > 1$.

More recently, several other real-world scenarios of DA have been introduced. For instance, Liu et al. (2019a) proposed wildly UDA, a realistic problem setting where predictors are forced to be trained with noisy labeled data from source domain and unlabeled data from target domain. Peng, Wu, and Ernst (2018) developed zero-shot domain adaptation, a setting where only task-irrelevant target-domain data is available during training. Moreover, Wang, He, and Katabi (2020) introduced continuously indexed domain adaptation, where the label space is continuously changing.

In this thesis, we concentrate on unsupervised domain adaptation (UDA) under closed-set, single-source, and single-target settings. UDA is one of the most challenging scenarios for domain adaptation, a problem setup where samples from the target domain are available, but none of them have been labelled. In what follows, we introduce the notations, a formal definition of UDA, and existing assumptions in UDA in Sec 2.2.1. Then, we provide a review of the prominent generalization bounds for domain adaptation in Sec. 2.2.2. The existing research on unsupervised deep domain adaptation applied to computer vision applications is presented in Sec. 2.2.3.

2.2 Literature Review

2.2.1 Preliminary

Notations We use $\mathcal{X}$ and $\mathcal{Y}$ to denote the input and output spaces respectively. Accordingly, we use $\mathbf{x}$, $\mathbf{y}$ as random variables from spaces $\mathcal{X}$, $\mathcal{Y}$.

Definition 2.3. **(*Classification*)** Classification is a machine learning task that aims to learn a function using labeled datapoints to map input samples to an output space $\mathcal{Y}$, defined by $h : \mathcal{X} \to \mathcal{Y}$, where function h (hypothesis) belongs to the set of all possible functions $\mathcal{H}$ called hypothesis space. Given a loss $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the error of a hypothesis h with respect to the true

labeling function $f : \mathcal{X} \to \mathcal{Y}$ under joint probability distribution $p(\mathbf{x}, \mathbf{y})$, known as the hypothesis risk, is defined as: $\varepsilon(h) = \varepsilon^l(h, f) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[l(h(\mathbf{x}), \mathbf{y})\big]$. If f and h are binary functions, then we let $\mathcal{Y} = \{0, 1\}$, and l be the zero-one loss, $l(h, f) = \mathbb{1}\big[h \neq f\big]$. Due to the access to only a finite number of samples, $\{x_i, y_i\}_{i=1}^N$, in practice, the expected value of hypothesis risk with respect to the joint distribution of data and labels is approximated by the sample average, called empirical risk $\widehat{\varepsilon}(h) = \frac{1}{N}\sum_{i=1}^N \big[l(h(\mathbf{x}_i), \mathbf{y}_i)\big]$.

Definition 2.4. **(*UDA*)** Given N_s labeled samples of source domain $\{(x_i, y_i)|$ $x_i \in \mathcal{X}_s, y_i \in \mathcal{Y}_s, i = 1, 2, ...N_s\}$, distributed according to density $p_s(\mathbf{x}, \mathbf{y})$, and unlabelled samples of target domain $\{(x_i)|x_i \in \mathcal{X}_t, i = 1, 2, ...N_t\}$, distributed according to density $p_t(\mathbf{x})$, UDA aims to transfer the knowledge learned from the source domain to the target domain. Formally, if we let $\widehat{\varepsilon_s}(h)$ to be the empirical source risk h, and similarly, we use $\varepsilon_t(h)$ and $\widehat{\varepsilon_t}(h)$ to mean the true risk and the empirical risk on the target domain respectively, the problem of domain adaptation can be stated as: under what conditions and by what algorithms can we guarantee that a small source empirical error $\widehat{\varepsilon_s}(h)$ implies a small true target (test) error $\varepsilon_t(h)$? The true target error can be defined by

$$\begin{aligned}\varepsilon_t(h) &= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_t(\mathbf{x},\mathbf{y})}\big[l(h(\mathbf{x}), \mathbf{y})\big] \\ &= \frac{1}{|\mathcal{Y}|}\sum_{\mathbf{y}\in\mathcal{Y}}\int_{\mathcal{X}}\left[l(h(\mathbf{x}), \mathbf{y})\frac{p_t(\mathbf{x}, \mathbf{y})}{p_s(\mathbf{x}, \mathbf{y})}p_s(\mathbf{x}, \mathbf{y})d\mathbf{x}\right].\end{aligned} \tag{2.16}$$

In traditional machine learning, we assume that $\frac{p_t(\mathbf{x},\mathbf{y})}{p_s(\mathbf{x},\mathbf{y})} = 1$, which implies that both training and testing data are drawn from the same distribution. Domain adaptation relaxes this assumption to some extent. However, still further assumptions should be considered to simplify $\frac{p_t(\mathbf{x},\mathbf{y})}{p_s(\mathbf{x},\mathbf{y})}$. Depending on sources of variations between source and target domains, the following assumptions are considered in the literature:

1. Covariate shift, when $p_s(\mathbf{y}|\mathbf{x}) = p_t(\mathbf{y}|\mathbf{x})$ for all $\mathbf{x}$, and $p_s(\mathbf{x}) \neq p_t(\mathbf{x})$.
2. Label shift, when $p_s(\mathbf{x}|\mathbf{y}) = p_t(\mathbf{x}|\mathbf{y})$ for all $\mathbf{y}$, and $p_s(\mathbf{y}) \neq p_t(\mathbf{y})$.
3. Concept drift, when $p_s(\mathbf{x}) = p_t(\mathbf{x})$, and $p_s(\mathbf{y}|\mathbf{x}) \neq p_t(\mathbf{y}|\mathbf{x})$.

Under *covariate shift* scenario, it is assumed that the shift between the source and target domains are merely caused by inconsistency in the feature space ($p_t(\mathbf{x}) \neq p_s(\mathbf{x})$). Importance Sampling is employed by Shimodaira (2000) to bridge the distributional gap via a weighting mechanism ($w(\mathbf{x}) = \frac{p_t(\mathbf{x})}{p_s(\mathbf{x})}$).

However, the shift between two domains with high dimensional data, such as texts or images, stems from non-overlapping supports, thus requiring unbounded weights. Ben-David et al. (2007) theoretically analyzed that the non-overlapping supports can be reconciled by learning invariant representations. This led to numerous algorithms to solve domain adaptation, which primarily aligns the source and target domains in the representation space.

2.2.2 Probability Divergences and Generalization bound

The prerequisite for a deep understanding and practical development of unsupervised domain adaptation algorithms is to consider the possibility of generalization across probability distributions. In this part, we provide the fundamental domain adaptation generalization bounds (GB) introduced in the literature. These bounds rely on the divergence measures between the probability distributions.

2.2.2.1 $\mathcal{L}_1$-distance-based GB

From a theoretical perspective, the problem of domain adaptation was initially studied by Ben-David et al. (2007). The authors first proposed $\mathcal{L}_1$ distance as a measure of divergence between two probability distributions.

Definition 2.5. Let $\mathcal{A}$ be a set of measurable subsets under the marginal probability distributions $\mathcal{D}_s$ and $\mathcal{D}_t$. The $\mathcal{L}_1$ distance or divergence between two domains can be defined as:

$$d(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{A \in \mathcal{A}} \Big| Pr_{\mathcal{D}_s}(A) - Pr_{\mathcal{D}_t}(A) \Big|. \tag{2.17}$$

Then based on the $\mathcal{L}_1$ distance, the first generalization bound was proposed by the same authors (Ben-David et al., 2007) as follows.

Theorem 2.1 (Ben-David et al., 2007). Given two domains $\mathcal{D}_s$ and $\mathcal{D}_t$ over $\mathcal{X} \times \mathcal{Y}$, and a hypothesis function h, the following holds.

$$\varepsilon_t(h) \leq \varepsilon_s(h) + d(\mathcal{D}_s, \mathcal{D}_t) + \min \left\{ \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_s} \Big[\Big| f_s(\mathbf{x}) - f_t(\mathbf{x}) \Big| \Big], \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_t} \Big[\Big| f_t(\mathbf{x}) - f_s(\mathbf{x}) \Big| \Big] \right\}, \tag{2.18}$$

where $f_s(\mathbf{x})$ and $f_t(\mathbf{x})$ are the source and target true labeling functions. The generalization upper bound is decomposed into three parts: the true source error, the empirical $\mathcal{L}_1$-distance, and the shift between labelling functions. However,

the tightness of this bound is infeasible to evaluate, as the $\mathcal{L}_1$ distance cannot be estimated from finite samples for arbitrary probability distributions.

2.2.2.2 $\mathcal{H}$-divergence-based GB

Definition 2.6. (Ben-David et al., 2010). Let us assume $\mathcal{D}_s$ and $\mathcal{D}_t$ to be the marginal distributions of source and target domains over the input space $\mathcal{X}$ respectively. Let $\mathcal{H}$ be a hypothesis class on $\mathcal{X}$ and denote by $I(h)$ the set for which $h \in \mathcal{H}$ is the characteristic function; that is, $\mathbf{x} \in I(h) \Leftrightarrow h(\mathbf{x}) = 1$. Then, the $\mathcal{H}$-divergence between $\mathcal{D}_s$ and $\mathcal{D}_t$ is defined as:

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{h \in \mathcal{H}} \left| Pr_{\mathcal{D}_s}(I(h)) - Pr_{\mathcal{D}_t}(I(h)) \right|. \tag{2.19}$$

An estimation of divergence from finite samples can be drawn using the following lemma.

Lemma 2.1. (Ben-David et al., 2010) Let $\mathcal{U}_s$ and $\mathcal{U}_t$ be sets of unlabelled samples of size m each, drawn from $\mathcal{D}_s$ and $\mathcal{D}_t$ respectively, and $\mathcal{H}$ be a hypothesis space of VC dimension d then for any $\delta \in (0, 1)$, the following holds with probability of at least $1 - \delta$

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) \leq \widehat{d}_{\mathcal{H}}(\mathcal{U}_s, \mathcal{U}_t) + 4 \sqrt{\frac{d \log(2\ m) + \log(\frac{2}{\delta})}{m}}, \tag{2.20}$$

where the estimated divergence between the unlabelled samples can be approximated with:

$$\widehat{d}_{\mathcal{H}}(\mathcal{U}_s, \mathcal{U}_t) = 2 \left(1 - \min_{h \in \mathcal{H}} \left[\frac{1}{m} \sum_{\mathbf{x} \in \{\mathbf{x} | h(\mathbf{x}) = 0\}} \mathcal{I}\Big[\mathbf{x} \in \mathcal{U}_s\Big] + \frac{1}{m} \sum_{\mathbf{x} \in \{\mathbf{x} | h(\mathbf{x}) = 1\}} \mathcal{I}\Big[\mathbf{x} \in \mathcal{U}_t\Big] \right] \right), \tag{2.21}$$

with $\mathcal{I}[\mathbf{x}]$ being the indicator function, which returns 1 if $\mathbf{x}$ is true, 0 otherwise. Ben-David et al. (2010) showed that the empirical $\mathcal{H}$-divergence given above is the error of the best classifier for the binary classification problem for the source and target samples.

2.2.2.3 $\mathcal{H}\triangle\mathcal{H}$-divergence-based GB

Definition 2.6. (Ben-David et al., 2010). Let us assume $\mathcal{D}_s$ and $\mathcal{D}_t$ to be the marginal distributions of source and target domains over the input space $\mathcal{X}$ respectively. Let $\mathcal{H}$ be a hypothesis class, and let $\mathcal{H}\triangle\mathcal{H}$ represents symmetric

difference hypothesis space defined as the following for $(h, h^{'}) \in \mathcal{H}^2$

$$r \in \mathcal{H}\triangle\mathcal{H} \Leftrightarrow r(\mathbf{x}) = h(\mathbf{x}) \oplus h^{'}(\mathbf{x}), \tag{2.22}$$

where $\oplus$ is *XOR* operation. The $\mathcal{H}\triangle\mathcal{H}$-divergence between two marginal probability distribution $\mathcal{D}_s$ and $\mathcal{D}_t$ is defined as follows:

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{(h,h')\in\mathcal{H}^2} \Big|\varepsilon_s(h, h') - \varepsilon_t(h, h')\Big|, \tag{2.23}$$

where $\varepsilon_s(h, h')$ and $\varepsilon_t(h, h')$ are the disagreement between two hypotheses h and h' on the source and target domains respectively.

Theorem 2.2. (Ben-David et al., 2010) Let $\mathcal{H}$ be a hypothesis space of *VC* dimension $VC(\mathcal{H})$. If $\mathcal{U}_s$ and $\mathcal{U}_t$ are unlabeled samples of size m each, which are drawn independently from $\mathcal{D}_s$ and $\mathcal{D}_t$ respectively, then for any $\delta \in (0, 1)$ with probability of at least $1 - \delta$ (over the choice of the samples), and for all $h \in \mathcal{H}$ we have

$$\varepsilon_t(h) \leq \varepsilon_s(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_s, \mathcal{U}_t) + 4\sqrt{\frac{2\ VC(\mathcal{H}) \log(2\ m) + \log(\frac{2}{\delta})}{m}} + \Psi(h^*), \tag{2.24}$$

where the optimal joint hypothesis h^* is defined as $h^* = \arg\min_{h\in\mathcal{H}} \varepsilon_t(h^*) + \varepsilon_s(h^*)$, and its corresponding combined error is $\Psi(h^*) = \varepsilon_t(h^*) + \varepsilon_s(h^*)$.

The presence of the trade-off between source risk, divergence, and capability to adapt is a very important phenomenon in domain adaptation. Indeed, it shows that the reduction in the divergence between the samples can be insufficient when there is no hypothesis that can achieve a low error on both the source and target samples.

2.2.2.4 $\widetilde{\mathcal{H}}$-divergence-based GB

Lemma 2.2. (Zhao et al., 2019) Let $\mathcal{H} \subseteq [0, 1]^{\mathcal{X}}$, $h, h^{'} \in \mathcal{H}$, $0 \leq t \leq 1$, and $\widetilde{\mathcal{H}} := \big\{\text{sgn}\big(\big|h(\mathbf{x}) - h^{'}(\mathbf{x})\big| - t\big)\big|\big\}$, where *sgn* denotes the sign function, then $\widetilde{\mathcal{H}}$-distance is defined as follows

$$d_{\widetilde{\mathcal{H}}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{(h,h')\in\mathcal{H}} \sup_{t\in[0,1]} \Big|Pr_{\mathcal{D}_s}\big(\big|h(\mathbf{x}) - h^{'}(\mathbf{x})\big| > t\big) - Pr_{\mathcal{D}_t}\big(\big|h(x) - h^{'}(\mathbf{x})\big| > t\big)\Big|. \tag{2.25}$$

Theorem 2.3. (Zhao et al., 2019) Let $\langle \mathcal{D}_s, f_s \rangle$, $\langle \mathcal{D}_t, f_t \rangle$ be the source and target true distributions. For any function class $\mathcal{H} \subseteq [0,1]^{\mathcal{X}}$, and $\forall h \in \mathcal{H}$, the target error can then be bounded by the source error $\varepsilon_s(h)$, the discrepancy between marginal distributions $d_{\widetilde{\mathcal{H}}}(\mathcal{D}_s, \mathcal{D}_t)$, and the distance between the optimal source and target labeling functions of $f_s : \mathcal{X} \to [0,1]$, and $f_t : \mathcal{X} \to [0,1]$ respectively, as in

$$\varepsilon_t(h) \leq \varepsilon_s(h) + d_{\widetilde{\mathcal{H}}}(\mathcal{D}_s, \mathcal{D}_t) + \min\left\{ \mathbb{E}_{\mathcal{D}_s}\Big[\Big|f_s - f_t\Big|\Big], \mathbb{E}_{\mathcal{D}_t}\Big[\Big|f_s - f_t\Big|\Big] \right\}. \tag{2.26}$$

The upper bound in Equation 2.26 no longer relies on Ψ, as in Equation 2.24. Zhao et al. (2019) also developed an information-theoretic lower bound for target error.

Theorem 2.4 (Zhao et al., 2019) Let $\mathbf{y} = f(\mathbf{x}) \in \{0,1\}$ be the labeling function, $\mathbf{y}' = h(g(\mathbf{x})) \in \{0,1\}$ be the prediction function, which is the predicted random variable of interest, suppose the Markov chain $\mathbf{x} \xrightarrow{g} \mathbf{z} \xrightarrow{h} \mathbf{y}'$ and condition $d_{\mathcal{JS}}(\mathcal{D}_s^{\mathbf{y}}, \mathcal{D}_t^{\mathbf{y}}) \geq d_{\mathcal{JS}}(\mathcal{D}_s^{\mathbf{z}}, \mathcal{D}_t^{\mathbf{z}})$ holds, then the following equation provides a lower bound on the joint source and the target error.

$$\underbrace{\varepsilon_s(h \circ g) + \varepsilon_t(h \circ g)}_{①} \geq \frac{1}{2}\Big(\underbrace{d_{\mathcal{JS}}(\mathcal{D}_s^{\mathbf{y}}, \mathcal{D}_t^{\mathbf{y}})}_{②} - \underbrace{d_{\mathcal{JS}}(\mathcal{D}_s^{\mathbf{z}}, \mathcal{D}_t^{\mathbf{z}})}_{③} \Big)^2. \tag{2.27}$$

The lower bound in Equation 2.27 provides us with a necessary condition on the success of any domain adaptation approach based on learning invariant representations: if the marginal label distributions are significantly different between source and target domains, that is, term ② is high, then minimizing term ③ together with the source error $\varepsilon_s(h \circ g)$, will only increase the target error $\varepsilon_s(h \circ g)$. In other words, if the term ② is high, in order to achieve a small joint error (term ①), the distribution divergence in representation space (term ③) has to be high as well, which means that the invariance goal needs to be relaxed.

Summary

In this section, we presented several theoretical results including those proposed by Ben-David et al. (2010) and Zhao et al. (2019) that establish the conditions under which a UDA problem can be solved efficiently. Following these theoretical insights, we design a new algorithm for UDA based on relaxing the excessive invariance.

2.2.3 Critical Review on UDA

The development of deep neural networks has enabled unprecedented performance in a wide variety of computer vision tasks. However, one of the major obstacles to deep learning, which prevented it from becoming a safely deployable machine learning algorithm, is its inability to generalize well to new domains. Even slight deviation from the training domain can give rise to significant performance deterioration (Ben-David et al., 2007). Technically, not only can the underlying issue stem either from the scarcity or unavailability of labeled data for the new domain, but also from the covariate, prior probability, or concept shift in data distribution (Zhang et al., 2019a). UDA aims to overcome these labelled data availability and distributional discrepancy problems by transferring knowledge from an available richly-labelled domain (i.e. source domain) to a target domain without labelled data. To tackle this, classes of algorithms under different assumptions, and based on well-studied theoretical bounds (Ben-David et al., 2007; Ben-David et al., 2010; Le et al., 2018; Zhang et al., 2019c; Zhao et al., 2019), have been proposed in the literature. In this section, given the extensive literature on UDA, we review researches that are most relevant to our thesis. Hence, we categorize distinct lines of research into different groups, and in each group, we highlight the most relevant papers. These include semi-supervision-based methods, distribution matching methods, intermediate representations, and architecture design. In addition, self-supervision-based methods (Ghifary et al., 2016; Sun et al., 2019), and causality-based methods (Magliacane et al., 2017), have been explored by researchers, which are beyond the scope of this thesis.

A. Semi-supervision-based Methods

Unsupervised domain adaptation (UDA) and Semi-Supervised Learning (SSL) are closely related; in both cases, we are given labeled and unlabeled data, with the core objective of learning a classifier capable of generalizing to the unlabeled data and unseen examples. However, in SSL, both the labeled and unlabeled data come from the same distribution, while in UDA, the target and source distributions differ. SSL methods, such as consistency regularization (Temporal Ensembling (Laine and Aila, 2016), Mean teachers (Tarvainen and Valpola, 2017), Virtual Adversarial Training (Miyato et al., 2018), Adversarial Dropout (Park et al., 2018), Interpolation Consistency Training (Zhang et al., 2017)), proxy-label methods (Pseudo-labeling (Arazo et al., 2020), Co-training (Blum and Mitchell, 1998), Tri-Training (Zhou and Li, 2005)), Graph-based

methods (e.g., Label propagation (Zhur and Ghahramani, 2002)), and entropy minimization methods (Grandvalet, Bengio, et al., 2005) demonstrated their effectiveness in domain adaptation. In this section, we highlight those papers that tackled UDA problem via SSL techniques without distribution matching techniques, which we also call ***non-invariant representation*** learning.

It has long been recognized that an ensemble of several neural networks generally produces better predictions than a single neural network does (Laine and Aila, 2016) . This idea has been exploited in unsupervised learning for generating predictions on unlabelled data, a technique called self-ensembling. In fact, self-ensembling is to use only a single model but running it under different operating conditions (such as inserted noise) that may produce different outputs; the unsupervised learning then aims for consistency between the self ensembles. This has been accomplished by either averaging over past predictions for each example, also known as Temporal Ensemble (Laine and Aila, 2016), or directly averaging over past network weights instead of predictions, also known as Mean Teacher (Tarvainen and Valpola, 2017). The latter strategy has been extended for domain adaptation problem (French, 2017). The input images augmented by various data augmentation techniques are passed through two networks with different mechanism of learning: a student network, which is trained with gradient descent, and a teacher network whose weights are an exponential moving average (EMA) of the student network's weights. At each step, the student network evaluates images from the source domain and computes derivatives via a task loss based on the ground truth. The unlabelled target domain images are used to compute the consistency loss by comparing predictions from both student and teacher models.

Pseudo-labelling can be seen as an equivalent implementation of entropy minimization that enables supervised training on unlabeled data. By iterative learning, the pseudo-labeling is expected to be gradually more accurate until convergence. Sener et al. (2016) proposed label learning by using k-nearest neighbors between unlabeled target samples and labelled source samples. This method jointly learns the transferable domain-specific representations and estimate the labels of the unsupervised data points. Saito, Ushiku, and Harada (2017) employed a target-specific classifier to extract discriminative target features by using pseudo-labelling, which is acquired confidently through two other classifiers on source and target representations.

Lee et al. (2019b) proposed drop to adapt (DTA), where a combination of SSL losses namely adversarial dropout loss (Miyato et al., 2018), virtual adversarial training loss (Tarvainen and Valpola, 2017), and entropy minimization loss (Grandvalet, Bengio, et al., 2005) have been used to extract the discriminative representation for unsupervised domain adaptation.

Critical point: in general, non-invariant representation methods learns discriminative representations, but they rely on domain-specific data augmentation strategies, in which case they may fail to generalize to the target domains far away from the source domain given that they do not consider the domain shift issue.

B. Distribution Matching Methods

Currently, the most prominent UDA methods are based on distributional matching (DM). DM has been applied either in the representation space, known as domain-invariant representation, or in the pixel space, which is called domain-mapping. DM methods are divided into three categories based on generative/discriminative settings. In what follows, we review the relevant papers in each category.

B.1. Domain-invariant Representation

A prevalent approach acting as a major component in numerous proposed algorithms for UDA is the estimation of ***domain-invariant representation***. Its main idea is to reduce the divergence between two domains to obtain domain-invariant distribution such that a predictor trained by source samples can be directly employed to the target data. Two domains can be invariant to the domain shift by learning features that have the same distribution regardless of their true underlying domains. The main hypothesis behind these methods is that such a common latent feature representation exists, and the label distributions do not differ significantly. Under this motivation, methods, such as statistical matching using kernelized training (Tzeng et al., 2014; Long et al., 2017; Sun and Saenko, 2016; Courty et al., 2017; Damodaran et al., 2018), and adversarial matching using adversarial training (Tzeng et al., 2014), have been proposed.

B.1.1. Statistical Matching

These approaches reduce the domain shift by minimizing some domain discrepancy metrics such as maximum mean discrepancy (MMD), correlation alignment loss, contrastive domain discrepancy (CDD), and the Wasserstein distance. In this stage we review statistical matching based deep domain adaptation methods.

Tzeng et al. (2014) proposed the deep domain confusion (DDC), where an adaptation layer derived from a pre-trained model is embedded into the last layers of a feature extractor and a confusion discrepancy loss based on maximum mean discrepancy is enforced on this layer across the source and target distributions. Tzeng et al. (2015) extended the previous work by introducing soft label distribution matching loss. Long et al. (2015) proposed to reduce the marginal distribution discrepancy across domains by transferring features of the top layers called task-specific layers to a Reproducing Kernel Hilbert Space (RKHS), where the the multi kernel MMD (MK-MMD) was utilized to match two domains onto each other. This idea was further developed by joint adaptation networks (JAN) (Long et al., 2017), which used a unified MMD called joint MMD (JMMD) to learn transferable features by matching the joint distribution of multiple domain-specific layers.

In previous approaches, the source classifier and target classifier are assumed to be the same. Long et al. (2016) relaxed the shared-hypothesis space assumption and related two classifiers from two domains by a residual function, which is a small perturbation function. Then, adaptation is enabled for the target classifier by learning this residual function and exploiting the entropy minimization that supports the low-density boundary assumption. The aforementioned MMD-based DA approaches failed to take the changes in prior class distributions into account. In fact, MMD is unable to compensate for class weight discrimination and results in reduced domain adaptation performance. To address this issue, a weighted model is proposed by Yan et al. (2017), where class specific auxiliary weights are implemented with the original MMD. In contrast to the methods mentioned above, Yan et al. (2017) proposed central moment discrepancy that minimizes the domain-specific features representations directly in the hidden activation space.

Correlation alignment (CORAL), designed by Sun and Saenko (2016), used

second-order statistical criteria (covariances) in loss function to reduce the discrepancies between two domains. The idea is similar to (DDC)(Tzeng et al., 2014), except that instead of MMD the CORAL loss is used to minimize discrepancy. Other distances for correlation alignment have been used. For instance, Zhang et al. (2018b) proposed to use a Euclidean distance in mapped correlation alignment (MCA), and Morerio, Cavazza, and Murino (2017) used geodesic distances. Contrastive domain discrepancy (CDD) proposed by Kang et al. (2019) exploited kernelized training, but looks at conditional distributions to incorporate label distribution. It proposes a contrastive adaptation network (CAN), which performs alternating optimization between adaptation of feature representations by backpropagation and updating the target labeling function by clustering for class-conditional alignment. When optimizing CDD, intra-class discrepancy is minimized while inter-class margin is maximized. The Wasserstein distance has been employed by Shen et al. (2017) to measure the distances between distributions. To align feature and label distributions with this distance, Courty et al. (2017) proposed the joint distribution optimal transport (JDOT). Damodaran et al. (2018) proposed DeepJDOT to incorporate this into a deep neural network.

Critical point. All the aforementioned statistical approaches are considered to be problem-specific and hard to design. Furthermore, minimizing non-parametric statistical distances, such as maximum mean discrepancy, fails to capture the structure of complex real-world distributions and restricts the notion of similarity to enable closed-form estimation.

B.1.2. Adversarial Matching

Inspired by the generative adversarial networks (GAN) (Goodfellow et al., 2014), adversarial training has been successfully employed to learn domain-invariant representation. In adversarial training strategy, the minimization of the domain discrepancy encourages domain confusion in the representation space, where the discriminator (a parameterized binary classifier) cannot distinguish whether the sample comes from the source or from the target domain. A general network setup of adversarial domain-invariant strategy for domain adaptation is illustrated in Figure 2.1.

Motivated to minimize the domain discrepancy measured by $\mathcal{H}$-divergence of (Ben-David et al., 2007), Ganin and Lempitsky (2014) proposed the first strategy of the domain-adversarial training of neural networks (DANN), where a

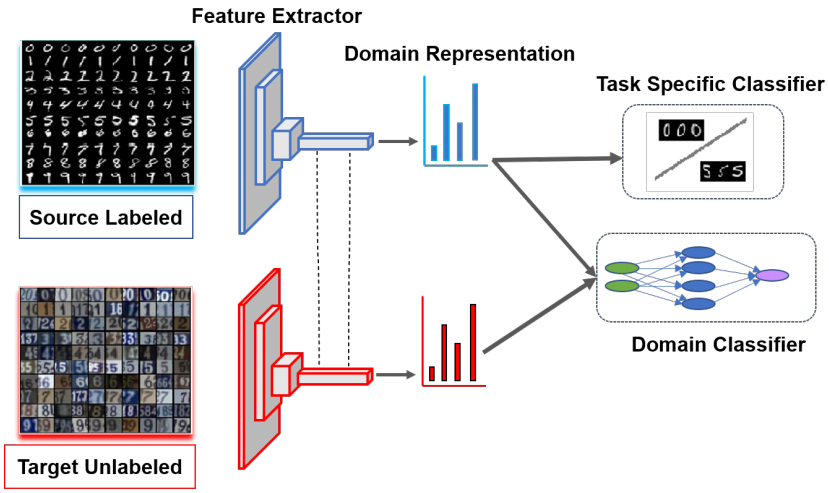


FIGURE 2.1. General network setup for adversarial domain-invariant representation learning. Dashed lines denote shared feature extractor.

binary classifier is adopted as the domain discriminator, and the domain divergence is minimized by learning representations of the two domains in an adversarial manner to the domain discriminator. Adversarial domain adaptation is achieved by adding an effective gradient reversal layer to the standard architecture, which reverses the discriminator's gradient during back-propagation by multiplying it with a negative scalar. Gradient reversal layer confirms that the representations over both source and target distributions are similarly generated (as the domain classifier will not be able to classify those representations), leading to the domain-invariant features. Tzeng et al. (2017) proposed adversarial discriminative domain adaptation (ADDA), where different from Ganin and Lempitsky (2014) and Tzeng et al. (2014), who used true minimax objective and domain confusion objective, respectively, ADDA utilized standard GAN loss (Goodfellow et al., 2014) to deal with the gradient vanishing problem. The discriminator tries to distinguish the source domain features from the target domain features whereas the feature extractor tries to confuse the discriminator to generate or extract domain invariant or domain independent features. After matching source and target representations, a task-specific classifier trained on the source distributions is applied directly to the target distributions. DANN uses a common feature extractor whereas ADDA uses two feature extractor to extract discriminative features for both domain and it solves the gradient vanishing problem. Volpi et al. (2018) improved ADDA by introducing data augmentation in the feature space where domain alignment is held as an ADDA objective.

Critical point. Drawbacks can be highlighted with the standard domain adversarial training. First, they suffer from the multi-modality bottleneck. Indeed, the discriminator can fail to capture multimodal structures of data distributions. Second, they make two domains closer to each other under domain-invariance constraint without considering the capacity of feature extraction function. Third, they are based on the assumption that the existence of one optimal hypothesis for two domains is guaranteed. Finally, the relationship between unlabeled samples and the decision boundary is ignored, which results in poor generalization of the target domain. The methods below have been proposed to address these issues.

- **Multimodal Alignment.** Motivated to capture the multimodal structures underlying data distributions when aligning domains, Long et al. (2018) developed a conditional domain adversarial network (CDAN), where a multilinear map function was employed to integrate the predictor's output and the feature representation to jointly learn the domain discriminator. Pei et al. (2018) proposed the multi-adversarial domain adaptation (MADA) approach for minimizing the domain discrepancy using multiple domain classifiers to address the mode collapse issue. In MADA, the soft pseudo-label of a target sample is used to determine how much this sample should be attended by different class-specific domain discriminators. Zhang et al. (2018a) proposed a collaborative adversarial network where multiple feature extractors and multiple domain discriminators are used to decrease the domain disparity.

- **Target-discriminative Representation.** The existing techniques in semi-supervised learning (SSL) algorithms have been implemented to improve the classifier for the target samples. These techniques capture target structures by entropy minimization (Sener et al., 2016; Long et al., 2016; French, 2017; Shu et al., 2018; Liang et al., 2018), pseudo-labelling (Saito, Ushiku, and Harada, 2017; Zhang et al., 2018a), consistency regularization (Saito et al., 2017; Saito et al., 2018; Kumar et al., 2018), and a combination of these methods (Kang et al., 2019; Chen et al., 2019b). The smoothness assumption has been applied on the classifier's prediction by penalizing inconsistent prediction between the perturbed version of the samples and original ones. To this end, Shu et al. (2018) relied on virtual adversarial training (VAT) (Miyato et al., 2018) to adjust the decision boundary against locally and randomly perturbed samples. To avoid overfitting to the unlabeled datapoints, the conditional entropy loss

has been used in combination with VAT loss. Dirt-T model (Shu et al., 2018) improved upon VADA by further penalizing a failure to meet the cluster assumption (Chapelle and Zien, 2005), that the data distribution tends to form separated clusters and that data points in the same cluster are more likely to share the same class label. Deng, Luo, and Zhu (2019) proposed cluster alignment with a teacher (CAT) where a teacher classifier is implemented to predict the cluster alignments for target samples.

- **Local Alignment.** From an adversarial distribution matching perspective, UDA methods can be categorized into two groups: (i) Global Alignment (GA) methods, for example, the aforementioned methods (Ganin et al., 2016; Shu et al., 2018; Pei et al., 2018) can be considered as GA methods, and (ii) Local Alignment (LA) methods (Saito et al., 2018; Lee et al., 2019a; Zhang et al., 2019b; Zhang et al., 2019c). GA methods tend to ignore the local class decision boundary information during adaptation, which leads to a sub-optimal performance on the target domain. Although several recently proposed GA methods try to benefit from cluster assumption to capture target-discriminative representation, they still suffer from a fundamental drawback. The domain discriminator enables matching the marginal feature distribution across domains by simply predicting the domain label, but it fails to take the class information into account. LA methods (Saito et al., 2018; Saito et al., 2017; Lee et al., 2019a) strive to address this issue by playing adversarial games between feature extractor and the disagreement of two classifiers on the target samples, as illustrated in Figure 2.2. Note that two task specific classifiers are initialized differently to acquire different classifiers from the beginning of training.

  More specifically, Maximum classifier discrepancy (MCD), proposed by Saito et al. (2018), aims to find the target samples that are distant from the source (outside the support of the source) by maximizing the $\mathcal{L}_1$-distance between the outputs of two task-specific classifiers employed as a discriminator, and minimizing this discrepancy to generate representations that are close to the source (inside the support of the source). In other words, the discrepancy between two domains is measured by the disagreement between two hypotheses. Indeed, it focuses on directly reshaping the target data regions that need to be reshaped. Adversarial dropout regularization (ADV) proposed by the same author (Saito et al., 2017) also tried to achieve discriminative target representation for adversarial domain adaptation via dropout regularization on discriminator. By doing

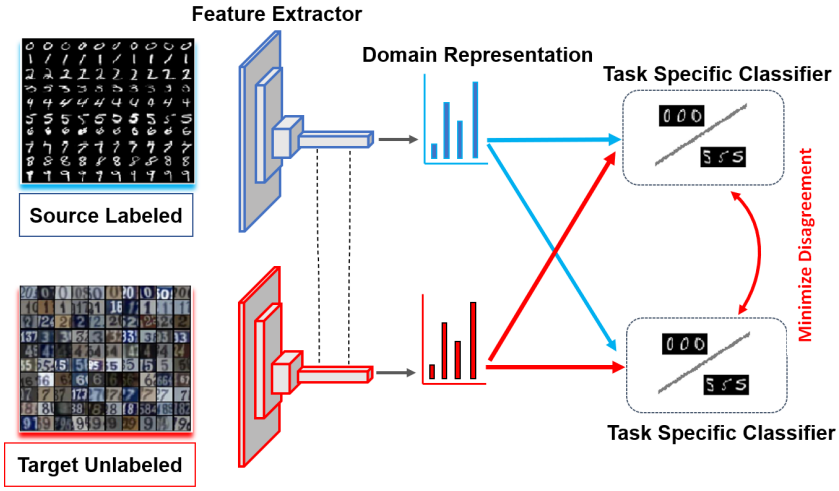


FIGURE 2.2. General network setup for adversarial local alignment. Dashed lines denote shared feature extractor.

so, two specific classifiers are created, and then the posterior discrepancy between the two classifiers is trained to be maximized while the generator is enforced to output discriminative features, which are assumed to be in high-density regions. Sliced Wasserstein discrepancy (SWD) proposed by Lee et al. (2019a) supplanted the $\mathcal{L}_1$-distance in (Saito et al., 2018) with the Wasserstein distance (Bonneel et al., 2015) to take advantage of its geometrical characterization. Zhang et al. (2019b) proposed SymNets where the domain confusion and the domain discrimination are stacked upon a classifier concatenated by two task-specific classifiers of source and target domain to facilitate the domain-level and category-level feature distribution alignment. Zhang et al. (2019c) proposed MDD where two classifiers are used asymmetrically to estimate conditional feature distributions with margin loss; Cicek and Soatto (2019) proposed a joint domain-class discriminator with the help of pseudo-labels instead of a binary discriminator to make any classifier agnostic to domains.

- **Other Methods.** Adversarial learning has been recently explored by other researchers (Wen et al., 2019; Zou et al., 2019; Chen et al., 2018; Chen et al., 2019a) to achieve domain invariant representations. Inspired by Arandjelovic et al. (2016), Wen et al. (2019) proposed to learn local features along with holistic distribution matching. Wang et al. (2019b) introduced an attention module with adversarial learning to minimize the negative transfer learning especially when the source and target images from the same category are very dissimilar. Zou et al. (2019) proposed consensus adversarial domain adaptation, where model setup has four steps

including training only with source data, training with source and target data and domain alignment through adversarial learning, fine-tuning in source domain with shared classifier, and testing in the target domain. Chen et al. (2018) introduced re-weighted adversarial unsupervised domain adaptation network to discriminatively align the source and target domains, and used optimal transport distance to tackle the issue of disjoint support. To alleviate mis-labeling issue in pseudo labeling technique, Chen et al. (2019a) proposed progressive feature alignment network where target samples with similarity scores higher than a threshold are selected for pseudo-labeling and this threshold is updated after each iteration of learning so that more unlabeled target samples can be selected.

B.2. Relaxed/Regularized Domain-invariant Representation

More recently, it was argued that invariant enforcement in the representation space is too restrictive and can substantially deteriorate its adaptability (minimal joint error of two domains) or discriminability (Wu et al., 2019; Johansson, Ranganath, and Sontag, 2019; Zhao et al., 2019; Arjovsky et al., 2019; Bouvier et al., 2019a). For instance, Wu et al. (2019) provided a theoretical analysis on learning invariant representation when there is an insufficient joint support, and showed that complete matching may increase target error. In *Target Shift* setting, where only the change in label distribution needs to be addressed, Zhao et al. (2019) showed that searching for invariance may damage adaptability. In the context of *Covariate Shift*, where the typical assumption is that source and target domains only differ in their feature distribution, Bouvier et al. (2019a) takes the risk of compression into account when enforcing invariance on representation, to control the adaptability. Bouvier et al. (2019b) proposed Hidden Covariate Shift which matches a reweighted source joint distribution and an estimated joint target distribution by imposing the invariance constraint on conditional label distribution given the data representation. In a general case of domain adaptation, Arjovsky et al. (2019) points out that it is not clear how invariant-representation can help aligning two domains.

Based on the aforementioned theoretical contributions, some recent works (Cao, Long, and Wang, 2018; Combes et al., 2020; Wu et al., 2019) systematically analyze the issue with exact distribution matching resulted from label distribution mismatch and proposed to either regularize or relax domain-invariance.

- **Regularized-invariance Representation.** Some other methods in UDA are focused on modifying the feature properties, namely transferability

(Chen et al., 2019c; Xu et al., 2019), and discriminability (Chen et al., 2019c; Xu et al., 2019; Cui et al., 2020a; Jin et al., 2020) along with the distribution matching. Chen et al. (2019c) proposed Spectral Batch Normalization (BSP), where the largest singular values of the feature matrix that are found to correspond to transferability, was penalized to enhance the features discriminability. Xu et al. (2019) attributed the nature of low transferability and model degradation to an extremely smaller norms of the target-domain features with respect to that of the source domain, and thus proposed parameter-free Adaptive Feature Norm (AFN) to enhance transferability by progressively equalising feature value ranges of two domains, which allows features with initially low ranges to impact the classification result more effectively. Cui et al. (2020a) proposed Batch Nuclear-norm Maximization (BNM) to simultaneously improves the discriminability and diversity of conditional prediction features. The nuclear-norm of prediction features' matrix is bounded by the Frobenius-norm of the matrix, which is a measure of prediction features' discriminability, and approximates the matrix rank, which refers to prediction features' diversity. Maximizing nuclear-norm ensures large Frobenius- norm of the batch matrix, leading to increased discriminability. Similarly, Jin et al. (2020) proposed Versatile Domain Adaptation (VDA), which enhances the diversity and discriminability of conditional prediction matrix through class confusion minimization.

- **Relaxed-invariance Representation**. More recently, two works tried to address the trade-off between transferablity and discriminability by relaxing domain invariance with weighted representations (Combes et al., 2020; Wu et al., 2019). However, they considered hypotheses in their methods that may fail to to be justified. For instance, Combes et al. (2020) proposed generalized label shift under the hypothesis that the probability of the latent space given the label in two domains is the same ($p_t(\mathbf{z}|\mathbf{y}) = p_s(\mathbf{z}|\mathbf{y})$), which ignores the intra-class feature distribution shift. Another method proposed by Wu et al. (2019), attempted to address the label distribution shift by asymmetrically relaxing the distribution alignment with hypothesis that the density ratios in representation space are upperbounded by a certain constant ($\frac{p_t(\mathbf{z})}{p_s(\mathbf{z})} \leq 1 + \beta, \beta > 0$), which does not take potential disjoint source-target supports into account. Therefore, learning both transferable and discriminable representations remains an open problem. Following this motivation, this thesis aims to introduce the

limitation of invariant representation learning in chapter 3, and propose efficient solution to relax excessive invaraince enforcement in chapter 4.

B.3. Domain Mapping

In comparison with the methods that align distributions in the feature space, several methods align distributions in pixel space by translating source domain to the target domain, and training the classifiers directly on the translated samples. This is accomplished through image-to-image (I2I) translation GAN. Liu and Tuzel (2016) proposed to train a pair of GANs called CoGAN on two domains of images. They use a GAN to generate corresponding images in multiple domains and then employ all but the last layer of the discriminator as a feature extraction function for classification. A recognized unsupervised I2I method is CycleGAN, developed by Zhu et al. (2017). CycleGAN overcomes the under-constrained nature of GAN, with a cycle-consistency enforcement. CycleGAN has been employed to perform UDA. Liu, Breuel, and Kautz (2017) coupled CoGAN with VAE (Kingma and Welling, 2013) to perform unsupervised I2I translation. A shared latent space between source and target domains is inferred to align the joint distributions of different domains.

However, such image-space adaptation methods cannot cope with large image sizes and large domain shifts, which means that they do not necessarily preserve the contents. To address these issues, Hoffman et al. (2017) proposed cycle-consistent adversarial domain adaptation (CyCADA), which performs adaptation at both the pixel-level and feature-level, enforces cycle-consistency, and uses a semantic loss to preserve semantics under a large domain shift. Yet, the advantage of the cycle-consistency loss is not demonstrated sufficiently, as this loss is only applied on pixel-space, which is suitable for low-level features. Also, this augmented semantic loss may fail to guarantee the semantic consistency at higher levels of deep representations.

C. Intermediate Representations

Regardless of the strategy to align the source and target distributions, UDA methods vary based on their choice of representation. Multiple UDA methods utilize intermediate representations (sub-spaces) between the source and target domains to reduce the domain shift. These methods avoid the domain-invariant representation and explore domain-specific features to minimize the domain mismatch through mapping one of the source or target domains to the other by making gradual alteration to the training distribution (Gong et al., 2012; Chopra,

Balakrishnan, and Gopalan, 2013; Fernando et al., 2013). The transferred representations smoothly bridge the gap between source and target domains, thereby facilitating the domain adaptation task. For instance, in the literature of shallow domain adaptation, Gong et al. (2012) regard two sub-spaces as two points on a Grassmann manifold, and find points on a geodesic path between them as a bridge between source and target sub spaces. Gopalan, Li, and Chellappa (2011) represent each domain as a subspace or covariance matrix, and then connect them on the corresponding manifold to model intermediate domains. This asymmetric transformation of source domain to target domain has recently led to numerous asymmetric UDA algorithms and is a promising approach, as higher classification accuracy has been achieved (Elhadji-Ille-Gado, Grall-Maes, and Kharouf, 2017; Thopalli et al., 2019). However, these methods cannot be easily applied to deep networks.

Recently, the idea of bridging the representations between source and target domains has been investigated in the field of domain adaptation (Gong et al., 2019; Liu et al., 2019b; Cui et al., 2020b). Gong et al. (2019) proposed domain flow for adaptation (DLOW) to generate multiple intermediate domains between source and target domains using normalizing flow to reduce the domain shift, as opposed to direct image-to-image translation (Zhu et al., 2017). In this method, the geodesic transfer properties are enforced by reconstructing input images. Gradually vanishing bridge (GVB) proposed by Cui et al. (2020b) explicitly reduces the domain discrepancy via a gradual transferring process undertaken on a geodesic domain flow along the data manifolds on both generator and discriminator without completely reconstructing domain-specific inputs. However, in these methods, input image reconstruction is imposed to enhance the geodesic transferability, where removing the domain-specific properties from domain-invariant features tends to be less accurate (Arjovsky et al., 2019). Liu et al. (2019b) introduced transferable adversarial training (TAT), where transferable examples are generated to fill the domain gap, and are augmented to the datasets. However, this method may fail to generalize to the target domains far away from the source domain due to the ignorance of domain shift.

D. Architecture Design

Another line of research in domain adaptation involves architectural design, and more specifically, the Batch Normalization (BN) layer (Ioffe and Szegedy, 2015) design, which is fundamental to deep learning models and also an essential part of domain adaptation. BN aims to stabilize the distribution of the minibatch

inputs to a network layer during training by setting the mean and variance of the distribution to be zero and one respectively. As with domain adaptation, since the normalization statistics are different across domains, per-domain batch normalization is proposed in the literature (Li et al., 2016; Carlucci et al., 2017; Roy et al., 2019; Wang et al., 2019a). Li et al. (2016) proposed Adaptive Batch Normalization (AdaBN) under the assumption that the domain-invariant information is stored in the neural net layer weights, while the domain-specific information is hidden in the statistics of the BN layer. AdaBN matches the source and target representations by using different mean and variance terms for the source and target domain when performing BN at test time. In other words, AdaBN leverages target statistics at test time to reduce the domain discrepancy. However, these statistics are excluded from the training procedure. In fact, the target data are not used to learn the network weights but only for adjusting the batch norm statistics during the test. Carlucci et al. (2017) proposed automatic domain alignment layers (AutoDIAL), which are embedded in different levels of the deep network before each BN layer to align the source and the target features. AutoDIAL learns to inject a suitable combination of the source and target features to BN layer at the training stage. Roy et al. (2019) proposed domain-specific whitening transform (DWT), where the source and target data distributions are aligned via their covariance matrices. Wang et al. (2019a) proposed Transferable Normalization (TransNorm), where the statistics of source and target data are calculated separately, while the channel transferability is computed simultaneously. The normalized features are then re-weighted by using a channel-wise distance operator, which is inversely proportional to the channel transferability.

Critical point. Batch normalization layer design can improve the performance of UDA methods. However, rare attention has been paid to the other aspects of architectural design, such as number of layers, and more importantly, the type of layers (e.g., convolutions, linear, or affine coupling layer), which are essential parts of domain adaptation. In this thesis, we show how affine coupling blocks can considerably improves UDA performance.

2.3 Chapter Summary

In this chapter we have first provided the readers with a background on unsupervised representation learning, transfer learning, and domain adaptation. In detail, we introduced three generative models, namely VAE, GAN, and NF,

which are widely used as the backbone of UDA models. We have then presented a clear definition of transfer learning and domain adaptation. In the second part of this chapter, we have provided a comprehensive literature review of UDA, where we mathematically defined UDA and described the required assumptions to solve it. We have then provided an overview of the existing theoretical bounds that have been proven for the domain adaptation problem. These theoretical guarantees provide crucial insight into empirical results. We have also managed to classify UDA methods related to this thesis into four categories (i.e. semi-supervision-based methods, distribution matching methods, intermediate representations, and architecture design), and summarized the representative papers in each category. In the second category, we critically explained the issues with the domain-invariant representation approach in dealing with UDA, and pointed out the solutions proposed by other researchers. Additionally, we placed and justified our research within this category.

Chapter 3

Representation Invariance and Theoretical Insights

Unsupervised domain adaptation (UDA) aims at enhancing the generalizability of the classification model learned from the labelled source domain to an unlabelled target domain. An established approach to UDA is to learn a domain-invariant representation via the alignment of the feature distributions of both domains. However, recent theoretical and empirical studies have revealed that complete source and target distribution matching fails to guarantee a small target error. To mitigate this issue and pave the way for designing an efficient algorithm for UDA, which will be introduced in the next chapter, in this chapter we first show how learning invariant representation may lead to an undesirable performance. Then we formalize a framework to seek a relaxed version of invariant representation learning. We also carefully characterize assumptions under which our framework is mathematically principled.

3.1 Introduction

We attempt to alleviate the problem of the domain-invariant representation learning by relaxing excessive invariance or regularizing the invariance mechanism in representation learning.

The contribution of this chapter is as follows: first, by leveraging deep generative models, including variational autoencoder and normalizing flow, from a probabilistic perspective, we formalize a lower bound on joint probability distribution of source and target domains as a unified framework for domain adaptation. MapFlow enables us (1) to model a more complex distribution for the target domain for which the density can be modeled when the source latent distribution is known, and (2) to model the relation between the two domains

rather than enforcing them to follow a simple and strict constraint (e.g. to be Gaussian distributed).

3.1.1 Limitations of Domain-Invariant Representations

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces respectively. $\mathcal{Z}$ is the representation space used by a feature transformation $g : \mathcal{X} \to \mathcal{Z}$. We also define an output labeling function $\varphi : \mathcal{Z} \to \mathcal{Y}$, and a composite predictive transformation $g \circ \varphi : \mathcal{X} \to \mathcal{Y}$. Let $\mathcal{H} = \{g \circ \varphi : g \in \mathcal{G}, \varphi \in \Phi\}$ be the hypothesis space, where Φ and $\mathcal{G}$ are considered to be the set of representations and predictive functions respectively. Given N_s labeled samples of source domain $\{(x_i, y_i)|x_i \in \mathcal{X}_s, y_i \in \mathcal{Y}_s, i = 1, 2, ...N_s\}$, with $(x, y) \sim p_s(\mathbf{x}, \mathbf{y})$, and N_t unlabelled samples of target domain $\{(x_i)|x_i \in \mathcal{X}_t, i = 1, 2, ...N_t\}$, with $x \sim p_t(\mathbf{x})$, UDA aims to transfer knowledge learned from the source domain to the target domain.

The error of a predictor φ with respect to the true labelling function f under distribution $\mathcal{D}$ with joint probability distribution $p(\mathbf{x}, \mathbf{y})$ is defined as: $\varepsilon(g, \varphi) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\big[\big|\varphi(g(\mathbf{x})) - f(\mathbf{x})\big|\big]$. Then for the target domain we have:

$$\varepsilon_t(h) = \int p_t(\mathbf{x})\big|\varphi(g(\mathbf{x})) - f(\mathbf{x})\big|d\mathbf{x}, \tag{3.1}$$

where $r(\mathbf{x}) = \big|\varphi(g(\mathbf{x})) - f(\mathbf{x})\big|$ is the risk for input $\mathbf{x}$. Following the change of variable rule ($\frac{p(\mathbf{x})}{p(\mathbf{z})} = \frac{d\mathbf{x}}{d\mathbf{z}}$), we then have

$$\varepsilon_t(h) = \int p_t(\mathbf{z})|\varphi(\mathbf{z}) - f_t(\mathbf{z})|d\mathbf{z} = \int p_t(\mathbf{z})r_t(\mathbf{z})d\mathbf{z}. \tag{3.2}$$

Similar to proof presented by Ben-David et al. (2010), $\varepsilon_t(h)$ can be simply redefined as follows:

$$\begin{aligned}
\varepsilon_t(h) &= \varepsilon_t(h) + \varepsilon_s(h) - \varepsilon_s(h) \\
&= \varepsilon_s(h) + \int p_t(\mathbf{z})|\varphi(\mathbf{z}) - f_t(\mathbf{z})|d\mathbf{z} - \int p_s(\mathbf{z})|\varphi(\mathbf{z}) - f_s(\mathbf{z})|d\mathbf{z} \\
&= \varepsilon_s(h) + \int p_t(\mathbf{z})r_t(\mathbf{z})d\mathbf{z} - \int p_s(\mathbf{z})r_s(\mathbf{z})d\mathbf{z}.
\end{aligned} \tag{3.3}$$

Let $\int p_t(\mathbf{z})r_s(\mathbf{z})d\mathbf{z}$ add to and subtract from Equation 3.3,

$$\begin{aligned}\varepsilon_t(h) = \varepsilon_s(h) + \int p_t(\mathbf{z})r_t(\mathbf{z})d\mathbf{z} - \int p_s(\mathbf{z})r_s(\mathbf{z})d\mathbf{z} \\ + \int p_t(\mathbf{z})r_s(\mathbf{z})d\mathbf{z} - \int p_t(\mathbf{z})r_s(\mathbf{z})d\mathbf{z},\end{aligned} \tag{3.4}$$

then we have:

$$\begin{aligned}\varepsilon_t(h) = \underbrace{\varepsilon_s(h)}_{①} + \underbrace{\int p_t(\mathbf{z})(r_t(\mathbf{z}) - r_s(\mathbf{z}))d\mathbf{z}}_{②} \\ + \underbrace{\int (p_t(\mathbf{z}) - p_s(\mathbf{z}))r_s(\mathbf{z})d\mathbf{z}}_{③}.\end{aligned} \tag{3.5}$$

The third term in Equation 3.5 is zero when $p_t(\mathbf{z}) = p_s(\mathbf{z})$, and the second term can become zero when the labeling function on representation space remains fixed between the source and target domains. Indeed, we have $r_t(\mathbf{z}) - r_s(\mathbf{z}) = |\varphi(\mathbf{z}) - f_t(\mathbf{z})| - |\varphi(\mathbf{z}) - f_s(\mathbf{z})| \leq |f_t(\mathbf{z}) - f_s(\mathbf{z})|$. However, as we do not have labels for the target domain, we have no control over the second term.

Wu et al. (2019) studied an upperbound to the third term of Equation 3.5, as follows:

$$\begin{aligned}\int (p_t(\mathbf{z}) - p_s(\mathbf{z}))r_s(\mathbf{z})d\mathbf{z} &= \int \left(\frac{p_t(\mathbf{z})}{p_s(\mathbf{z})} - 1\right)p_s(\mathbf{z})r_s(\mathbf{z})d\mathbf{z} \\ &\leq \left(sup_{\mathbf{z}\in\mathcal{Z}}\frac{p_t(\mathbf{z})}{p_s(\mathbf{z})} - 1\right)\varepsilon_s(h)\end{aligned} \tag{3.6}$$

This upperbound shows that if $\varepsilon_s(h) = 0$, then the condition $p_t(\mathbf{z}) = p_s(\mathbf{z})$ is no longer needed to make the third term in Equation 3.5 equal to zero. Note that in domain-invariant representation learning we assume that the ratio $\frac{p_t(\mathbf{z})}{p_s(\mathbf{z})}$ is equal to 1. In addition, Zhao et al. (2019) provided a counter example that shows if there is a label distribution mismatch, the condition of $\big(\varepsilon_s(h) = 0$, and $p_t(\mathbf{z}) = p_s(\mathbf{z})\big)$, leads to a positive error for target domain. Therefore, this equality enforcement $(\frac{p_t(\mathbf{z})}{p_s(\mathbf{z})} = 1)$ may deteriorate the adaptability. As a result, we suggest to relax this equality by finding a relationship between $p_t(\mathbf{z})$ and $p_s(\mathbf{z})$, or regularizing the invariant feature learning that will be explained in the next chapter.

3.2 A General Relaxed-Invariance Framework

The relationship between source and target domains can be captured by learning a transformation from source to target domain in the feature space. Instead of enforcing equality in the density of source and target representations, which is a major cause for inadaptability (Bouvier et al., 2019a), finding a relationship between source and target domains in representation space by an invertible neural network (INN) can be a solution. Accounting for the fact that a representation space may be more suitable for the target domain than it is for the source domain, invertible networks do not rely on strict domain invariance for better feature learning in two domains, enabling us to explicitly learn the target latent distribution. Explicit latent distribution modeling has been explored for UDA (Liu, Breuel, and Kautz, 2017; Grover et al., 2019; Zhu et al., 2019) and Semi-supervised domain adaptation (Pérez-Carrasco et al., 2019), where they model source and target latent distribution as predefined parametric distributions. However, different from those methods, normalizing flow (NF), which is a specific type of INN with an easily computable determinant of the Jacobian, can be employed to model the likelihood of complex target latent distribution.

In this section, we propose MapFlow, a general framework to relax domain-invariance. MapFlow framework (MFF) relies on normalizing flow to learn a bijective, non-linear transformation between the encoded target distribution and a flexible latent prior induced directly from the source latent space by variational inference. MFF is able to learn domain-specific knowledge by efficiently regularizing the Jacobian. To clarify, the maximization of the determinant of Jacobian helps to alleviate the distributional divergence, by establishing a geometrical relationship between the source and target representations. In addition, despite adversarial domain adaptation that may fail to achieve multimodal alignment, MFF can capture and preserve the multimodal structure of target latent space, which is suitable for having a discriminative mapping or alignment.

3.2.1 Framework for Joint Distribution

The learning of a joint distribution of source and target images has been studied and applied for domain adaptation (Liu, Breuel, and Kautz, 2017). However, those methods have a few limitations. First of all, they assume shared latent space or cycle-consistency, which are both rather restrictive, as they impose strict constraints while modeling complex distributions in the latent space. Secondly, they fail to achieve multimodal alignment. Thirdly, these methods

utilize adversarial training in representation spaces, which can be challenging due mainly to its unstable training dynamics. To tackle these limitations, a general framework is presented to infer the joint distribution from the marginal ones without any additional assumption on the structure of the joint distribution. In this framework, we generalize the relationship between source and target representations by using an invertible neural network, through which the distribution of the target representation can be modelled without enforcing a strict constraint. We formulate the lower bound on the joint probability distribution over data, which can be leveraged for the following multi-task learning objectives: 1) image translation between two domains, 2) sampling, and 3) classification.

We define a joint probability distribution over image samples and associated labels on both the source and target domains as follows: $p_\psi(\mathbf{x}_t, \mathbf{x}_s, \mathbf{y}_t, \mathbf{y}_s)$. Assuming the conditional independence between $\mathbf{y}_t$ and $\mathbf{x}_s$ given $\mathbf{x}_t$, and also the conditional independence between $\mathbf{y}_s$ and $\mathbf{x}_t$ given $\mathbf{x}_s$, the joint distribution can be factorized under the chain rule as follows:

$$p_\psi(\mathbf{x}_t, \mathbf{x}_s, \mathbf{y}_t, \mathbf{y}_s) = p_\gamma(\mathbf{x}_t, \mathbf{x}_s) p_\beta(\mathbf{y}_s|\mathbf{x}_s) p_\alpha(\mathbf{y}_t|\mathbf{y}_s, \mathbf{x}_t), \tag{3.7}$$

where $\psi = \{\gamma, \beta, \alpha\}$ represents the model parameters. The third term in Equation 3.7 can be interpreted as the probability of the model on target samples, the second term is the classification model on source samples, and the first term is the joint probability distribution over data samples, which can be defined as follows, by considering $\mathbf{z}_t$ and $\mathbf{z}_s$ as the latent variables to model the source and target distributions:

$$p_\gamma(\mathbf{x}_t, \mathbf{x}_s) = \int p_\theta(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) p_\eta(\mathbf{z}_t, \mathbf{z}_s) d\mathbf{z}_t d\mathbf{z}_s, \tag{3.8}$$

where finding the maximum likelihood of such joint distribution is generally intractable. Thus, we leverage variational inference for jointly modeling distributions. We assume joint variational posterior as $q_\phi(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)$, then the joint log-evidence lower bound (ELBO) can be derived as follows:

$$\begin{aligned}\log p_\gamma(\mathbf{x}_t, \mathbf{x}_s) \geq \mathbb{E}_{q_\phi(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)}\big[\log p_\theta(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s)\big] + \mathbb{E}_{q_\phi(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)}\big[\log p_\eta(\mathbf{z}_t, \mathbf{z}_s)\big] \\ - \mathbb{E}_{q_\phi(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)}\big[\log q_\phi(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)\big],\end{aligned} \tag{3.9}$$

where the first expectation term is a reconstruction error, the second one refers to the joint prior distribution, and the third expectation term minimizes the entropy of variational posterior. The reconstruction term can be factorized $p_\theta(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) = p_\theta(\mathbf{x}_t|\mathbf{z}_t)p_\theta(\mathbf{x}_s|\mathbf{z}_s)$ by assuming the conditional independence between $\mathbf{x}_t$ and $\mathbf{x}_s$ given $\mathbf{z}_t$, and the conditional independence between $\mathbf{x}_s$ and $\mathbf{z}_t$, given $\mathbf{z}_s$. To simplify the third term on the right hand side (RHS) of Equation 3.9, we formulate a factorized variational posterior of the form $q_\phi(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s) = q_{\phi_t}(\mathbf{z}_t|\mathbf{x}_t)q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)$, which is consistent with the conditional independence assumption between latent space of one domain and the input space of the other. Also, we define $\mathbf{z}_t = f(\mathbf{z}_s)$, which leads to factorization of joint prior as $p_\eta(\mathbf{z}_t, \mathbf{z}_s) = p_{\eta_t}(\mathbf{z}_t|\mathbf{z}_s)p_{\eta_s}(\mathbf{z}_s)$. Taking all these terms into account, and using the chain rule along with Equation 3.9, we can derive the final ELBO loss as follows (the further details about mathematical derivation of this loss can be found in the Appendix A.1):

$$\begin{aligned}\mathcal{L}_\gamma(\theta, \phi_s, \phi_t, \eta_s, \eta_t) = \lambda_{tr}\mathbb{E}_{q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)}\Big[\mathbb{E}_{p_{\eta_t}(\mathbf{z}_t|\mathbf{z}_s)}\big[\log p_\theta(\mathbf{x}_t|\mathbf{z}_t)\big]\Big] \\ + \lambda_{sr}\mathbb{E}_{q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)}\big[\log p_\theta(\mathbf{x}_s|\mathbf{z}_s)\big] \\ + \lambda_{kl}\mathbb{E}_{q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)}\big[\log p_{\eta_s}(\mathbf{z}_s) - \log q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)\big] \\ - \lambda_f\mathbb{E}_{q_{\phi_t}(\mathbf{z}_t|\mathbf{x}_t)}\left[\log p(f^{-1}(\mathbf{z}_t)) - \log\left|\det\frac{\partial f^{-1}}{\partial \mathbf{z}_t}\right|\right],\end{aligned} \tag{3.10}$$

where $\lambda = (\lambda_{sr}, \lambda_{tr}, \lambda_{kl}, \lambda_f)$ are regularization parameters. An illustration of our general framework is provided in Figure 3.1. It consists of one feature extractor (encoder) $g_s(\mathbf{x}_s; \phi_s)$ to learn posterior distribution for the source domain. We rely on variational inference (VI) to find an approximation $g_s(\mathbf{x}_s; \phi_s) = q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)$ for the true latent posterior distribution $p_\theta(\mathbf{z}_s|\mathbf{x}_s)$, which is parametrized by a deep neural network with parameters ϕ_s. Therefore, the representation space of the source domain is forced to be Gaussian with distribution $\mathcal{N}(\mathbf{z}_s|\mu_{\phi_s}(\mathbf{x}_s), \sigma^2_{\phi_s}(\mathbf{x}_s))$, which can be used as a prior to model target representation.

For the target domain, on the other hand, an invertible neural network constructed by affine coupling layers, which facilitates to compute the Jacobian $J = \frac{\partial f^{-1}}{\partial \mathbf{z}_t}$, has been utilized to estimate the density of target encoded samples $g_t(\mathbf{x}_t; \phi_t) = q_{\phi_t}(\mathbf{z}_t|\mathbf{x}_t)$.

Let $\mathbf{z}_s$ with dimension d be the encoded latent variable for unit Gaussian distribution $p(\mathbf{z}_s)$ and let $\mathbf{z}_t \in \mathcal{Z}_t$ be an observation from an unknown target

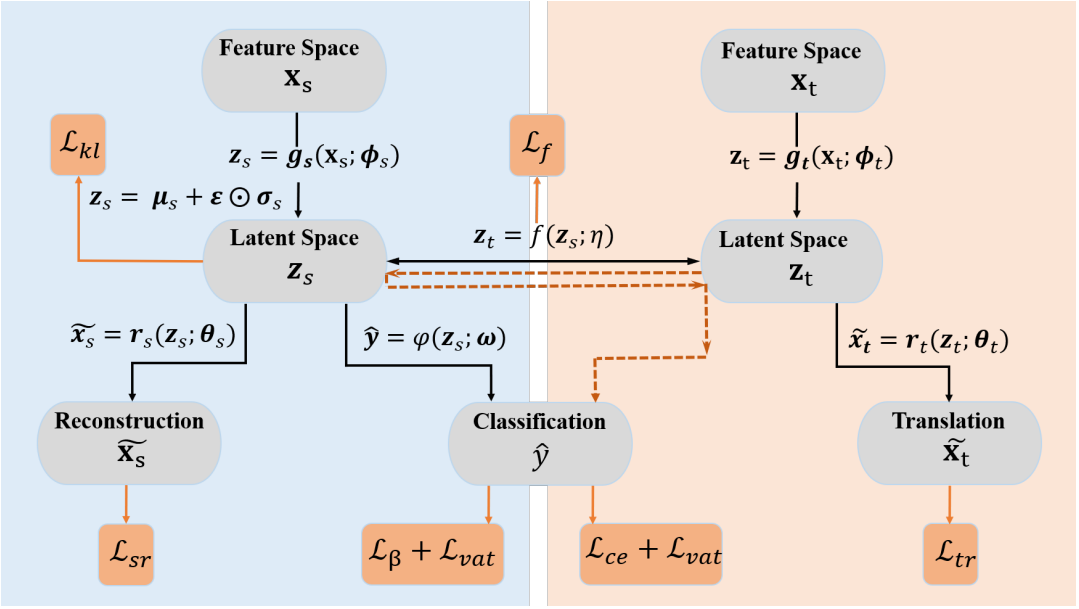


FIGURE 3.1. illustration of proposed unified framework for UDA

distribution $\mathbf{z}_t \sim p(\mathbf{z}_t)$. Given $f_\eta : \mathbf{z}_s \to \mathbf{z}_t$, we define a model $p_\theta(\mathbf{z}_t)$ with parameters θ on $\mathcal{Z}_t$, and we can compute the negative log likelihood (NLL) of $\mathbf{z}_t$ by the change of variable formula. For a single unlabeled target datapoint, the unsupervised objective can be derived as follows:

$$-\log p_{\eta_t}(\mathbf{z}_t) = \mathcal{L}_f(f_{\eta_t}(\mathbf{z}_t)) = -\Big(\log p_{\eta_s}(f_{\eta_t}^{-1}(\mathbf{z}_t)) + \log \Big| \det(\frac{\partial f_{\eta_t}^{-1}(\mathbf{z}_t)}{\partial \mathbf{z}_t}) \Big| \Big), \tag{3.11}$$

where p_{η_s} is the prior distribution for the source domain. The minimization of this loss helps to generate a mapping of each unlabeled target sample into the corresponding embedding space.

$\mathcal{L}_\gamma$ in Equation 3.10 has five terms including target reconstruction, source reconstruction, a prior term for source domain, which can be learned with another invertible network, entropy of source dataset, and a mapping objective from target to source. The learning of such a complex objective requires us to regularize each term with different weight. The second term in Equation 3.10 is a predictive function on source datasets.

Assuming that $p_\theta(\mathbf{z}_s|\mathbf{x}_s)$ can be approximated by the variational posterior $q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)$, we have:

$$p_\beta(\mathbf{y}_s|\mathbf{x}_s) = \int p_\omega(\mathbf{y}_s|\mathbf{z}_s) p_\theta(\mathbf{z}_s|\mathbf{x}_s) d\mathbf{z}_s \approx \mathbb{E}_{q_{\phi_s}(\mathbf{z}_s|\mathbf{x}_s)}[p_\omega(\mathbf{y}_s|\mathbf{z}_s)]. \tag{3.12}$$

The predictive function $\varphi_\omega : \mathcal{Z}_s \to \mathcal{Y}_s$ enforces separability between classes,

$$\mathcal{L}_\beta(\omega; \mathbf{z}_s) = -\mathbb{E}_{\mathbf{z}_s \sim q_{\phi_s}(\mathbf{z}_s|,\mathbf{x}_s)}[y_s^T \ln \varphi_\omega(\mathbf{z}_s)]. \tag{3.13}$$

Since we have no labels for the target domain, to learn a discriminative target representation, we follow (Shu et al., 2018; Kumar et al., 2018). We apply low-density and smoothness assumptions by assuming a conditional entropy (CE) minimization and virtual adversarial training (VAT).

$$\mathcal{L}_{ce}(\mathbf{z}_t; \omega) = -\mathbb{E}_{\mathbf{z}_t \sim q_{\phi_t}(\mathbf{z}_t|\mathbf{x}_t)}[\varphi_\omega(\mathbf{z}_t)^T \ln \varphi_\omega(\mathbf{z}_t)] \tag{3.14}$$

$$\mathcal{L}_{vat}(\mathbf{z}_t; \omega) = \mathbb{E}_{\mathbf{z}_t \sim q_{\phi_t}(\mathbf{z}_t|\mathbf{x}_t)} \big[\max_{\|r\| \leq \epsilon} D_{KL}(\varphi_\omega(\mathbf{z}_t) || \varphi_\omega(\mathbf{z}_t + r)) \big]. \tag{3.15}$$

While the conditional entropy minimization (Equation 3.14) forces the predictor to be confident on the unlabeled target data by pushing the decision boundaries away from the target data, VAT loss (Equation 3.15) enforces prediction consistency within the neighborhood of training samples. Note that VAT can be applied on both or either of the source and target distributions.

The overall objective of our proposed MFF to be minimized is given by:

$$\min_{\theta,\phi_s,\phi_t,\eta_s,\eta_t,\omega} \quad \mathcal{L}_\gamma(\theta, \phi_s, \phi_t, \eta_s, \eta_t) + \lambda_s \mathcal{L}_\beta(\omega; \mathbf{z}_s) + \lambda_t(\mathcal{L}_{ce}(\mathbf{z}_t; \omega) + \mathcal{L}_{vat}(\mathbf{z}_t; \omega)), \tag{3.16}$$

where $\mu = (\theta, \phi_s, \phi_t, \eta_s, \eta_t, \omega)$ are all parameters to be learned, and $\lambda = (\lambda_{sr}, \lambda_{tr}, \lambda_{kl}, \lambda_f, \lambda_s, \lambda_t)$ are regularization parameters.

3.3 Chapter Summary

In this chapter, we have analyzed the limitations of invariant-representation learning for UDA. We argued that the prevalent approach for UDA relies on a strict enforcement of the invariant representation for the underlying distributions, which can be too restrictive. Thus, we propose a general framework to relax invariance enforcement in representation space by using normalizing flow. Normalizing flow can be used to map the latent space of one domain to the latent space of the other domain.

Chapter 4

RIDA: Relaxed-Invariant Distribution Alignment for Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) aims at enhancing the generalizability of the classification model learned from the labelled source domain to an unlabelled target domain. An established approach to UDA is via domain-adversarial training, which has shown promising results to learn a domain-invariant representation. This is typically achieved by using a divergence measure to match the distribution of the representations of the source and target in a latent space. One major issue with this approach is that while one can match the distribution of the target and source, the individual classes are unidentifiable, i.e. the conditional distributions do not match. While this is conventionally presumed to be the down-stream classifier's task, the trade-off between the invariant distribution learning and the conditional classifier's loss is not trivial and requires specific architectural choices or data augmentation tricks. In this chapter we propose a simple, yet effective, solution to add an invertible function between the invariant embedding space and the classifier to learn a bijective transformation to facilitate classification. Intuitively if the invariant space is good enough for classification, this transformation is an identity map leading to the same performance as the baseline. However, when domain-invariant space is not conditionally identifiable, our transformation ensures each sample is mapped to a point that is easily separable for the down-stream classifier. Empirical results demonstrate the superior performance of our proposed algorithm compared to the relevant baselines without resorting to complicated data augmentations or other tricks. The method is easy to implement and generalizable to many neural network architectures.

4.1 Introduction

Unsupervised domain adaptation (UDA) aims at transferring discriminative features learned from labelled source domain to unlabelled target domain, with the difficulty of addressing distributional shift. To achieve a successful domain adaption, both transferability and discriminability in the feature learning should be guaranteed.

To learn discriminative features for the target domain, several methods, which we call **non-invariant representation methods**, focused exclusively on exploiting the techniques originally proposed in semi-supervised learning such as conditional entropy minimization (Prabhu et al., 2020), proxy labels (Saito, Ushiku, and Harada, 2017), consistency regularization (Liu et al., 2019b), and a combination of them (Lee et al., 2019b; Deng, Luo, and Zhu, 2019). However, these methods are unable to explicitly transfer the learned features, have no theoretical guarantee, and may not be applicable in realistic scenarios where large distribution gap between domains should be handled.

Feature transferability is most studied and predominantly enhanced by distribution matching in feature space, known as **domain-invariant representation** learning. Invariant representations have been achieved via moment matching (Sun and Saenko, 2016) and adversarial training (Ganin et al., 2016). One major issue with domain invariant representation learning is that it merely matches the distribution of the source and target domain in feature space without considering the discriminability of target features. Prior works address this by (i) utilizing two task specific classifiers and measuring the disagreement between their outputs on target samples as discrepancy between two domains (Saito et al., 2018; Lee et al., 2019a; Zhang et al., 2019b), (ii) coupling the non-invariant representation approaches with domain-invariant representation methods (Shu et al., 2018; Chen et al., 2020; Jiang et al., 2020), and (iii) regularizing the norm of invariant features (Chen et al., 2019c; Xu et al., 2019; Jin et al., 2020).

Nevertheless, more recently, it was argued that excessive invariance substantially deteriorate the discriminability (Wu et al., 2019; Johansson, Ranganath, and Sontag, 2019; Zhao et al., 2019). Indeed, the transferability of features in domain-invariance learning is strengthened at the expense of their discriminability. To tackle this trade-off, researchers attempted to relax the domain invariance with weighted representations (Combes et al., 2020; Wu et al., 2019).

However, they considered hypotheses in their methods that may fail to be justified. For instance, Combes et al. (2020) proposed generalized label shift under the hypothesis that the probability of the latent space given the label in two domains are the same ($p_t(\mathbf{z}|\mathbf{y}) = p_s(\mathbf{z}|\mathbf{y})$), which ignores the intra-class feature distribution shift. Wu et al. (2019) attempted to address the label distribution shift by asymmetrically relaxing the distribution matching with hypothesis that the density ratios in representation space are upperbounded by a certain constant ($\frac{p_t(\mathbf{z})}{p_s(\mathbf{z})} \leq 1+\beta, \beta > 0$), which does not take potential disjoint source-target supports into account.

Following this line of study and motivation, we propose to regularize the strict invariance between source and target domains by meaningfully re-weighting the invariant features. In fact, we empirically found that adding an invertible function between embedding function $g_\theta : \mathcal{X} \to \mathcal{Z}$ and embedding classifier $\varphi_\theta : \mathcal{Z} \to \mathcal{C}$ helps to mitigate the trade-off between invariant representation and discriminability. With that we ensure (1) the distribution of the transformed space is as similar as possible to the invariant embedding space, and (2) the features learnt are not degenerate since the transformation is bijective. In this approach, we believe that not all representation spaces should be reweighted equally. Therefore, by utilizing the invertible function, we automatically adjust the aligned representation weights.

In this chapter, we propose RIDA, a simple, yet effective methodology to tackle UDA. RIDA relies on normalizing flow to learn a bijective, non-linear transformation between the invariant embedding space and the embedding classifier. Avoiding any data augmentation techniques and consistency losses, RIDA method is able to relax the domain-invariant representation by regularizing the determinant of Jacobian. To elucidate, the maximization of the determinant of Jacobian helps to alleviate the enforcement on distributional divergence minimization, by establishing a geometrical relationship between invariant embedding space and the input for the classifier.

The main contributions of this chapter are as follows:

- We propose RIDA, which uses normalizing flow to map the invariant features to a space that is separable for the classifier.
- RIDA outperforms state-of-the-art results on several public UDA benchmarks.

4.2 Preliminary

4.2.1 Notation and Problem Definition

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output space, respectively. $\mathcal{Z}$ is the representation space generated from $\mathcal{X}$ by a feature transformation $g : \mathcal{X} \to \mathcal{Z}$. Accordingly, we use $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ as random variables from spaces $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, and let lower-case random variables x, y, and z denote the corresponding sample values respectively. We also define an output labeling function $\varphi : \mathcal{Z} \to \mathcal{Y}$ and a composite predictive transformation $g \circ \varphi$. Given N_s labeled samples of source domain $\{(x_i, y_i)|x_i \in \mathcal{X}_s, y_i \in \mathcal{Y}_s, i = 1, 2, ...N_s\}$, with $(x, y) \sim p_s(\mathbf{x}, \mathbf{y})$, and unlabelled samples of target domain $\{(x_i)|x_i \in \mathcal{X}_t, i = 1, 2, ...N_t\}$, with $x \sim p_t(\mathbf{x})$, UDA aims to transfer the predictive knowledge learned from the source domain to the target domain.

4.2.2 Normalizing Flow for transformation

The normalizing flow (Dinh, Sohl-Dickstein, and Bengio, 2016) is a likelihood-based generative model defined as an invertible function $f : \mathcal{X} \to \mathcal{Z}$ that maps the observed space $\mathcal{X}$ to the latent space $\mathcal{Z}$. The distribution of the observed variable can be modeled by applying a chain of invertible transformations, which is composed of a sequence of invertible functions $f^{-1} : \mathcal{Z} \to \mathcal{X}$ on random latent variables with known distribution $\mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z})$. Based on the change of variables formula, the negative log-likelihood of a single datapoint $\mathbf{x}$ can be computed as follows:

$$-\log\ p_{\mathcal{X}}(\mathbf{x}) = -\left(\log p_{\mathcal{Z}}(\mathbf{z}) + \log\left|\det(\frac{\partial f}{\partial \mathbf{x}})\right|\right), \tag{4.1}$$

where the scalar value $\log\left|\det(\frac{\partial f}{\partial \mathbf{x}})\right|$ represents the expansion or contraction of volume. The mapping $f(\mathbf{x}; \eta)$ is characterized by a deep neural network with an architecture that is carefully designed to ensure the invertibility and efficient computation of log-determinants, and a set of parameters η that can be optimized. We follow (Kingma and Dhariwal, 2018) and adopt the three main components of their model, that is, actnorm, random permutation, and affine coupling layers, to form the flow function used in our model.

4.3 Limitations and Insights

We let $\mathcal{D}_s$ denote the joint distribution of source domain over input x and one-hot label y, and let X_s denote the marginal input distribution. Similarly, we

define $(\mathcal{D}_t, X_t)$ for the target domain. The learning theory of UDA was proposed by Ben-David (Ben-David et al., 2010), and is summarized in Theorem 1.

Theorem 1 (Ben-David et al., 2010) *Let* $\mathcal{H} = \{\varphi \circ g : \varphi \in \Phi, g \in \mathcal{G}\}$ *be the hypothesis space, where* Φ *and* $\mathcal{G}$ *are considered to be the set of representations and predictive functions respectively, and let* $\varepsilon(h)$ *be the risk for* $h \in \mathcal{H}$*, and* $\varepsilon(h, h')$ *be the risk for* $(h, h') \in \mathcal{H}^2$.

$$\varepsilon_t(g \circ \varphi) \leq \underbrace{\varepsilon_s(g \circ \varphi)}_{①} + \underbrace{\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(X_s, X_t)}_{②} + \underbrace{\Psi(h)}_{③}, \tag{4.2}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}$ in second term denotes $\mathcal{H}\Delta\mathcal{H}$ distance between source and target domains, $\Psi(h)$ is the shared error of the ideal joint hypothesis, and

$$d_{\mathcal{H}\Delta\mathcal{H}}(X_s, X_t) = 2 \sup_{(h,h') \in \mathcal{H}^2} |\varepsilon_s(h, h') - \varepsilon_t(h, h')| \tag{4.3}$$

$$\Psi(h) = \inf_{h \in \mathcal{H}} \varepsilon_s(h) + \varepsilon_t(h). \tag{4.4}$$

Motivated by this theory, domain adversarial training (Ganin et al., 2016) minimizes a weighted combination of two objectives. The first objective has to do with learning feature discriminability in the source domain. This objective is trained with the labeled source data using the cross-entropy loss:

$$\mathcal{L}_{\mathbf{y}}(\theta; \mathcal{D}_s) = -\mathbb{E}_{\mathbf{x},\mathbf{y} \sim \mathcal{D}_s} \Big[\mathbf{y}^T \ln \varphi_\theta(g(\mathbf{x})) \Big]. \tag{4.5}$$

Along with the feature learning in the source domain, the next objective is to learn transferable features by minimizing the divergence between source and target representations, denoted by the loss

$$\mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t) = \sup_D \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_s} \Big[\ln D(g_\theta(\mathbf{x})) \Big] + \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_t} \Big[\ln(1 - D(g_\theta(\mathbf{x})) \Big], \tag{4.6}$$

where $D : \mathcal{Z} \to \{0, 1\}$ is domain discriminator. Our adversarial domain adaptation minimizes the following objective:

$$\min_\theta \mathcal{L}_{\mathbf{y}}(\theta; \mathcal{D}_s) + \beta \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t), \tag{4.7}$$

where β is a weighting factor. In representation learning for UDA, minimizing the distance between source and target domain, $d_{\mathcal{H}\Delta\mathcal{H}}(X_s, X_t)$, can be empirically achieved by minimizing the loss $\mathcal{L}_d$, which enhances the transferability of

features. However, separating domain-invariant information from the domain-specific information without any control over features or sufficient knowledge, may potentially deteriorate the discriminability, as original feature distributions are exposed to distortion. Discriminability can be defined as the capacity of clustering in the feature manifold. For UDA, we seek a good discriminability in both source and target domains simultaneously. As we typically employ a shared feature extractor for two domains, improving discriminability is equivalent to pursuing a better ideal joint hypothesis $\Psi(h)$, as defined in Equation 4.2. Nevertheless, due to the lack of labels in target domain, we have only been able to minimize the supervised loss of source domain, which consequently leads the model to be biased toward source-learned discriminative features. Hence, considering the limitations of domain adversarial training, we impose additional constraints to achieve more reliable domain adaptation.

4.4 Relaxed-invariance Approach

We relax invariant features by adding an invertible network between embedding space and embedding classifier, and then minimizing the determinant of input-output Jacobian of this network, which leads to increased discriminability. An illustration of our proposed idea is provided in Figure 4.1. As shown in the Figure, our model consists of a shared feature extractor $g : \mathcal{X} \to \mathcal{Z}$, a shared invertible network $f : \mathcal{Z} \to \mathcal{Z}_m$, which maps the latent space $\mathcal{Z}$ to a new space $(\mathcal{Z}_m)$, a shared classifier $\varphi : \mathcal{Z}_m \to \mathcal{C}$, and a discriminator D. We consider a composition of theses networks as classifier $h_\theta = g \circ f \circ \varphi$, parametrized by θ. Based on the Equation 4.1, we have:

$$\log p_{\mathcal{Z}}(\mathbf{z}) = \log p_{\mathcal{Z}_m}(f(\mathbf{z})) + \log \left| \det(\frac{\partial f}{\partial \mathbf{z}}) \right|. \tag{4.8}$$

Additionally, for the classifier to exploit unlabeled data, we apply conditional entropy minimization on the unlabeled data, which is a well-known regularizer in semi-supervised learning (Chapelle and Zien, 2005). This loss forces the decision boundaries not to be in the high-density region, causing the classifier to learn more discriminative features.

$$\mathcal{L}_e(\theta; \mathcal{D}_t) = -\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_t} \Big[h_\theta(\mathbf{x})^T \ln h_\theta(\mathbf{x}) \Big]. \tag{4.9}$$

The determinant of the Jacobian, $\det(\frac{\partial f}{\partial \mathbf{z}})$ at a given point $\mathbf{z}$, describes the behavior of the differentiable function f near that point. The absolute value of the

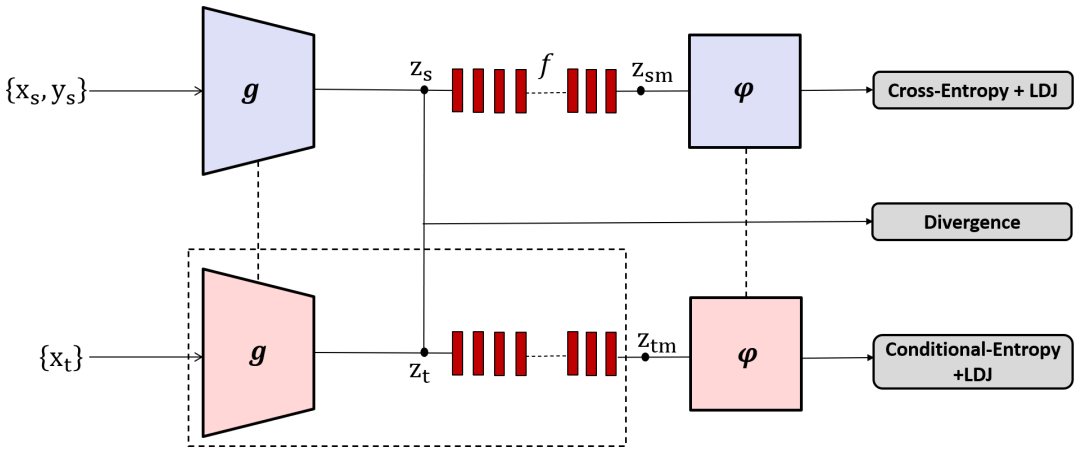


FIGURE 4.1. The network structure of proposed RIDA: invertible network f between embedding space and classifier helps to regularizes the invariant representation by penalizing the determinant of Jacobian. LDJ denotes log determinant of Jacobian.

Jacobian determinant at point $\mathbf{z}$, $|\det(\frac{\partial f}{\partial \mathbf{z}})|$, is a measure of local volume change in the neighborhood of that point. In other words, the expansion and contraction of volume, determines how much the function f is distorting the domain locally. Note that the objective of normalizing flows is to maximize likelihood $\log p_{\mathcal{Z}}(\mathbf{z})$ in Equation 4.8. Therefore, $|\det(\frac{\partial f}{\partial \mathbf{z}})|$ has an expansive effect on the volume. However, we minimize this term, which correlates with generalization, to keep the distribution of transformed space, $p_{\mathcal{Z}_m}(\mathbf{z}_m)$, as similar as possible to the distribution of invariant space, $p_{\mathcal{Z}}(\mathbf{z})$, as follows:

$$\mathcal{L}_m(\theta; \mathcal{D}_t) = \log \left| \det(\frac{\partial f}{\partial \mathbf{z}}) \right|. \tag{4.10}$$

We will apply this loss both on the source and the target training data. The invertible network can (1) ensure that the learnt features are not degenerate, 2) geometrically reshape the invariant embedding space by meaningfully scaling the features. Intuitively, weighting the embedding space via the determinant of Jacobian renders the features more robust to distortion caused by adversarial invariant learning.

Indeed, we maximize the absolute value of the Jacobian determinant of the inverse function, $|\det(\frac{\partial f^{-1}}{\partial \mathbf{z}})|$. This is inversely equal to the production of all the singular values of the Jacobian matrix, that is

$$\log \left| \det(\frac{\partial f^{-1}}{\partial \mathbf{z}}) \right| = \log \prod_{i=1}^{N} (\frac{1}{\sigma_i}) = \sum_{i=1}^{N} \log(\frac{1}{\sigma_i}), \tag{4.11}$$

where σ is the singular value, and N is the dimension of squared Jacobian matrix. In addition, we know that the Frobenius norm of the Jacobian matrix of f^{-1}, can be written as

$$\left|\left|\frac{\partial f^{-1}}{\partial \mathbf{z}}\right|\right|_F^2 = \sum_{i=1}^{N}(\frac{1}{{\sigma_i}^2}). \tag{4.12}$$

Based on Equation 4.11, and Equation 4.13, and the fact that, $\log\sigma < \sigma^2$, a lower bound and upper bound can be found for the Frobenius norm of the Jacobian matrix of f^{-1}, as follows:

$$\log\left|\det(\frac{\partial f^{-1}}{\partial \mathbf{z}})\right| < \left|\left|\frac{\partial f^{-1}}{\partial \mathbf{z}}\right|\right|_F^2 \leq \frac{1}{\sigma_{max}^2}, \tag{4.13}$$

where the σ_{max} denotes the largest singular value. Although maximizing the lower bound on the Frobenius norm of the Jacobian of f^{-1}, does not necessarily mean the minimization of the Frobenius norm of Jacobian matrix, we empirically found that this occurred when we added a Gaussian noise to the input of invertible network. As such, minimizing the determinant of Jacobian ensures small Frobenius norm of the Jacobian matrix, where the Frobenius norm of Jacobian matrix estimates the sensitivity of output to an input perturbation as follows:

$$\begin{aligned}\mathbb{E}_{\Delta x}\left[||f(\mathbf{x}) - f(\mathbf{x}+\Delta\mathbf{x})||_2^2\right] \approx \mathbb{E}_{\Delta\mathbf{x}}\left[||J(\mathbf{x})\Delta\mathbf{x}||_2^2\right] = \mathbb{E}_{\Delta\mathbf{x}}\left[\sum_i\left(\sum_j J_{ij}\mathbf{x}_j\right)^2\right] \\ = \sum ij J_{ij}^2 E_{\Delta\mathbf{x}}\left[\mathbf{x}_j^2\right] = \varepsilon||J(\mathbf{x})||_F^2,\end{aligned} \tag{4.14}$$

where $\Delta\mathbf{x} \sim \mathcal{N}(0, \varepsilon I)$ denotes a small Gaussian perturbation. Accordingly, the loss in Equation 4.11, leads to the features that are more robust to distortion. This will essentially control the adaptability or discriminability, allowing us to minimize the target error. We will prove this claim in the empirical study.

The overall objective function of the proposed RIDA model is defined by:

$$\min_\theta \mathcal{L}_y(\theta;\mathcal{D}_s) + \beta_d\mathcal{L}_d(\theta;\mathcal{D}_s,\mathcal{D}_t) + \beta_t\mathcal{L}_m(\theta;\mathcal{D}_t) + \beta_s\mathcal{L}_m(\theta;\mathcal{D}_s) + \beta_{ce}\mathcal{L}_e(\theta;\mathcal{D}_t), \tag{4.15}$$

where $(\beta_d, \beta_t, \beta_s, \beta_{ce})$ are the hyper-parameters that need to be estimated.

4.5 Experiments

In this section, we first present the experimental setup, then we provide details of the implementation of our model, followed by the results, where we compared our model with the SOTA methods in UDA, and an ablation study of the method.

4.5.1 Setup

Data Sets

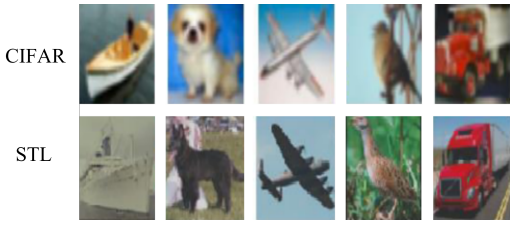
To demonstrate the performance of our proposed method, we present our model evaluation on three commonly used digit datasets for UDA: MNIST (LeCun, 1998), SVHN (Netzer et al., 2011), and USPS (Le Cun et al., 1990). For general object classification tasks, we rely on CIFAR-10 (Krizhevsky, Hinton, et al., 2009), STL-10 (Coates, Ng, and Lee, 2011), and office-31 (Saenko et al., 2010). Additionally, we evaluate our model to adaptation task on large-scale dataset. In particular, we test on VisDA-2017 (Peng et al., 2017) for image classification task. Figure 4.2 illustrates the sample images of aforementioned datasets.

USPS→MNIST. Modified National Institute of Standards and Technology (MNIST) is a binary handwritten digit dataset consisting of 70,000 samples split into 60,000 training samples and 10,000 test samples with a size of 28×28. Also, USPS dataset is handwritten digits scanned and segmented from envelopes by the U.S. Postal Service. USPS images are centered, normalized and gray scaled with 16×16 pixel. It has a training set of 7,291 images and 2,007 test images with various types of font styles. Moreover, in this adaptation task, the dimension of MNIST is reduced to 16×16 to match the dimension of USPS.

MNIST↔SVHN. In this adaptation task, the distributional shift is escalated. Whereas MNIST is composed of black and white handwritten digits, SVHN comprises a collection of colored, street house numbers.

CIFAR-10↔STL-10. Both STL-10 and CIFAR-10 datasets are equally distributed in 10 classes, but they contain nine overlapping classes. Following (Shu et al., 2018; Lee et al., 2019b), we redefine the adaptation task as a 9-class classification problem by removing the non-overlapping classes. Moreover, we reduce the dimension of STL from 96×96 to 32×32 to match the dimension of CIFAR10.

(a) The Digit datasets



(b) The CIFAR-10 and STL datasets

(c) Office-31

(d) VisDA-2017

FIGURE 4.2. Sample images of each dataset; a) Digit Datasets, b) CIFAR-10 and STL datasets, c) Office-31 dataset images of three domains d) VisDA-2017 dataset images of synthetic and real domains.

Office-31. It is a widely used dataset for unsupervised domain adaptation, which is comprised of 4, 110 images in 31 categories collected from three different domains: Amazon (A) from amazon.com, Webcam (W) taken by web camera, and DSLR (D) taken by the digital SLR camera.

VisDA-2017. It is a large-scale dataset with 12 classes for challenging UDA problem of adapting from synthetic images to real-world images. This dataset is comprised of 152,397 synthetic 2D images rendered from 3D images as the source domain. The target domain is composed of 55,388 real images taken from the MS-COCO dataset (Lin et al., 2014).

Baselines

We primarily compare our proposed RIDA with three baselines: ALDA (Chen et al., 2020), MDD+Implicit (Jiang et al., 2020), and VADA (Shu et al., 2018). We also show the results of several other recently proposed UDA models for comparison including Maximum Classifier Discrepancy (MCD) (Saito et al., 2018), Joint Adaptation Network (JAN) (Long et al., 2017), Self-Ensembling(S-En) (French, 2017), and Conditional Domain Adversarial Networks (CDAN) (Long et al., 2018). For fair comparison, the results are reported from the original papers if available. For all the experiments, we will report the results in terms of accuracy for each domain shift, repeating the experiments 3 times and averaging the results.

4.5.2 Implementation

Architecture

In order to make fair comparisons, for digits and CIFAR10/STL datasets, we adopt the architectural components including the classifier network, the feature extractor, and the discriminator used in DIRT-T (Shu et al., 2018). Similarly, we use a small architecture for the digits UDA tasks, and a larger architecture for UDA experiments between CIFAR-10 and STL-10. For office-31 and VisDA 2017 datasets, we employ ResNet-50 (He et al., 2016), which is pre-trained on ImageNet (Russakovsky et al., 2015), as the feature extractor. The discriminator network is composed of two fully connected layers with dropout (Ganin et al., 2016).

Note that our architecture is slightly different as we include an invertible feature transform to the classifier network; however, the invertible network only adds a small parameter overhead on the shared feature extractor and classifier (less than 4%). For the invertible network applied on latent variables, we use Glow architecture (Kingma and Dhariwal, 2018) with 4 affine coupling blocks, where each block contains 3 fully connected layers each with 256 or 512 hidden units depending on the dataset. The details of architectural components are presented in Appendix A.2.

Training Settings and Hyper-parameters

For digits and CIFAR10/STL datasests, we implement adversarial training via alternating updates (Shu et al., 2018), and train the model using Adam optimizer (Kingma and Ba, 2014) with learning rate 10^{-3} decaying by a factor of 2 after 200 epochs.

For office-31 and VisDA-2017 datasets, we follow (Chen et al., 2020), which adopted gradient reversal layer (Ganin and Lempitsky, 2014) to optimize discriminator, and follow all the protocols including optimizer, and learning rate strategy. We optimize the model using Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9 and an adjusted earning rate $\eta_p = \eta_0(1 + \alpha q)\gamma$, where $\eta_0 = 0.01$, $\alpha = 10$, $\gamma = 0.75$, and q is the training progress linearly decreasing from 1 to 0. Note that we set the learning rates of the classifier and discriminator to be 10 times that of the generator.

As for hyper-parameters $(\beta_d, \beta_t, \beta_s, \beta_{ce})$, we tune the values for each dataset using cross validation. We observed that the extensive hyper-parameter tuning is not required to obtain the top-performance results. A full list of the hyper-parameter settings is provided in Appendix A.3.

4.5.3 Results

Table 4.1 summarizes the results of the average accuracy (%) on the standard classification benchmarks for UDA such as digits, CIFAR-10, and STL data sets, compared with SOTA methods. For fair comparison, we resize all images to $32 \times 32 \times 3$ (except in case of adaptation from USPS to MNIST) and apply instance normalization (Shu et al., 2018) to input images. Note that all results are achieved without applying any data augmentation. Below, we present a brief analysis of the results in Table 4.1.

USPS→MNIST: although USPS contains smaller training set than MNIST, domain discrepancy between these two datasets is relatively small, and we could achieve a high performance in USPS → MNIST.

MNIST↔SVHN: for the adaptation task SVHN → MNIST, we modify the dimension of MNIST to 32×32 of SVHN, with three channels. This adaptation problem is easily solved When the proposed RIDA is applied. Our method could demonstrate a performance similar to the SOTA DTA (Lee et al., 2019b)

Source **Target**	USPS MNIST	MNIST SVHN	SVHN MNIST	CIFAR10 STL10	STL10 CIFAR10
DANN (Ganin et al., 2016)	-	35.7	71.1	-	-
UNIT (Liu, Breuel, and Kautz, 2017)	93.6	-	90.5	-	-
CYCADA (Hoffman et al., 2017)	96.5	-	90.4	-	-
Π-model (aug) (French, 2017)	97.3	71.4	92.0	76.3	64.2
CDAN (Long et al., 2018)	-	-	-	-	-
MCD (Saito et al., 2018)	94.1	-	96.2	-	-
VADA (Shu et al., 2018)	-	73.3	97.9	80.0	73.5
Dirt-T (Shu et al., 2018)	-	76.5	**99.4**	-	75.3
AlignFlow (Grover et al., 2019)	96.7	-	91.0	-	-
DTA (Lee et al., 2019b)	**99.1**	-	99.2	82.6	72.8
ALDA (Chen et al., 2020)	98.4	-	98.7	-	-
Source Only VADA (Shu et al., 2018)	-	40.9	82.4	77.0	62.6
Source Only Ours	78.4	42.2	82.1	76.7	63.1
RIDA(**Ours**)	**99.1**	**81.3**	**99.4**	**82.8**	**75.9**

TABLE 4.1. Test accuracy (%) on standard domain adaptation benchmarks. The model directly uses classifier trained on the source. Baseline numbers are taken from the cited works.

on MNIST. The reverse problem, the adaptation task MNIST → SVHN, can be regarded as the most challenging case in digit datasets, as MNIST has a considerably lower dimensionality than SVHN. Experiments show that RIDA could achieve state-of-the-art results on this adaptation task. On average, RIDA achieved **4.8**% improvements compared with the method of DIRT-T (Shu et al., 2018). The improvement shows the importance of relaxed invariant representation.

CIFAR-10↔STL-10: in both adaptation directions, results in Table 4.1 show that RIDA is slightly better than the SOTA, which we believe is due to the relatively smaller training set for STL and the existing imbalance between two datasets. It is important to note that we obtained this result without any SSL techniques and gradient refinement mechanisms as employed by Shu et al. (2018) and Lee et al. (2019b). Figure 4.3 illustrates the validation learning curve of the target domain for the adaptation task CIFAR10 → STL.

The results in Table 4.2 show again the superiority of our approach compared to other recently proposed methods on Office-31 datasets. We evaluate RIDA across six UDA tasks: A → W, W → D, D → W, A → D, D → A, and W → A. Our method surpasses the baselines in 3 out of 6 pairs of adaptation tasks for Office-31. Figure 4.4 shows the learning curve of the target domain

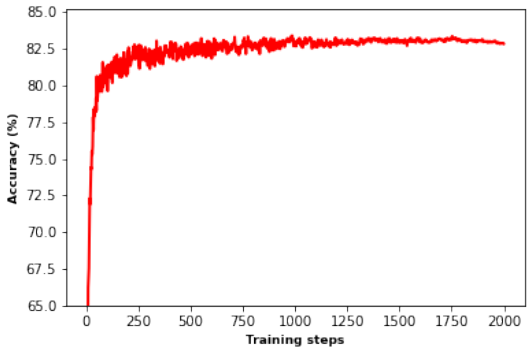


FIGURE 4.3. Learning curve of the target domain for the adaptation task CIFAR10 → STL

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
DANN (Ganin et al., 2016)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA (Tzeng et al., 2017)	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN (Long et al., 2017)	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MADA (Pei et al., 2018)	90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
MCD (Saito et al., 2018)	88.6±0.2	98.5±0.1	**100.0 ± .0**	92.2±0.2	69.5±0.1	69.7±0.3	86.5
CDAN (Long et al., 2018)	94.1±0.1	98.6±0.1	**100.0 ± .0**	92.9±0.2	71.0±0.3	69.3±0.3	87.7
MDD (Zhang et al., 2019c)	94.5±0.3	98.4±0.1	**100.0 ± .0**	93.5±0.2	74.6±0.3	72.2±0.1	88.9
ALDA (Chen et al., 2020)	95.6±0.5	97.7±0.1	**100.0 ± 0.0**	94.0±0.4	72.2±0.4	72.5±0.2	88.7
MDD + Implicit (Jiang et al., 2020)	90.3±0.2	**98.7 ± 0.1**	99.8±.0	92.1±0.5	**75.3 ± 0.2**	74.9±0.3	88.8
RIDA (**Ours**)	**96.4 ± 0.2**	97.8±0.2	99.8±0.1	**94.5 ± 0.3**	72.7±0.1	**75.9 ± 0.1**	**89.4**

TABLE 4.2. Test Accuracy (%) on Office-31 adaptation tasks for unsupervised domain adaptation (ResNet-50).

accuracy for A → W task. We further demonstrate the generalization ability

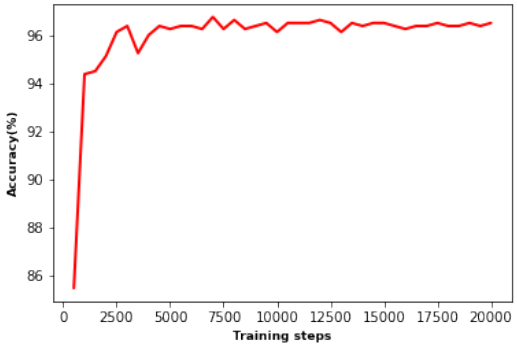


FIGURE 4.4. Learning curve of the target domain for the adaptation task A → W (ResNet-50)

of the proposed method by conducting additional experiments on VisDA-2017. In our experiments, we observed a gain of **0.4** points over the baseline (Chen

et al., 2020), confirming the flexibility of RIDA and its applicability across UDA tasks. The SOTA results with ResNet-50 are reported in Table 4.3.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg
CDAN (Long et al., 2018)	-	-	-	-	-	-	-	-	-	-	-	-	70.0
MCD (Saito et al., 2018)	90.3	62.6	**84.8**	71.7	85.9	72.9	93.7	71.9	86.8	**79.1**	81.6	14.3	74.6
MDD (Zhang et al., 2019c)	-	-	-	-	-	-	-	-	-	-	-	-	74.6
MDD+IA (Jiang et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	75.8
DTA (Lee et al., 2019b)	**93.1**	**70.5**	83.8	**87.0**	92.3	48.3	91.9	86.4	93.1	71.0	82.0	15.3	76.2
ALDA (Chen et al., 2020)	87.0	61.3	78.7	67.9	83.7	89.4	89.5	71.0	**95.4**	71.9	89.6	33.1	76.5
RIDA (**Ours**)	88.2	61.2	78.1	70.1	84.3	**90.2**	91.1	71.7	94.7	72.7	**89.9**	31.6	**76.9**

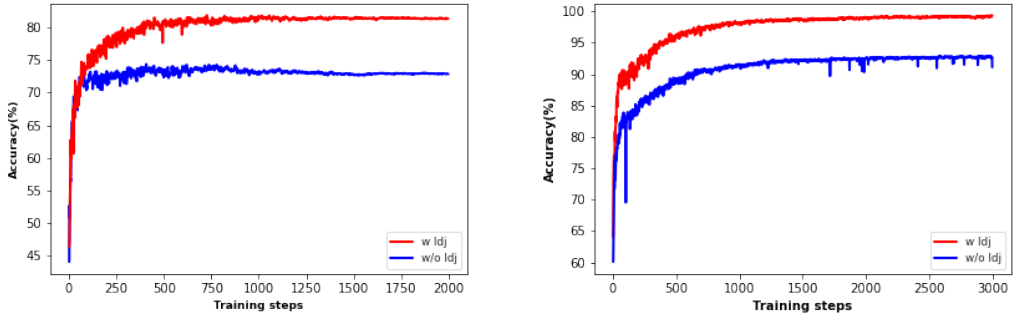
TABLE 4.3. Test Accuracy (%) on VisDA-2017 for unsupervised domain adaptation (ResNet-50).

4.5.4 Ablation Studies

To examine the relative contribution of augmented invertible network in RIDA, we conduct several ablations on the adaptation tasks presented in Table 4.1, with and without the log-determinant of Jacobian term (E.q 4.8). The results are reported in Table 4.4, where "no-ldj" subscript denotes the removal of the log determinant of Jacobian component. Also, Figure 4.5 illustrates the model behavior for SVHN $\leftrightarrow$ MNIST task with and without the log-determinant of Jacobian. We observe that only adding invertible network with affine coupling layers between embedding space and classifier ($\text{RIDA}_{\text{no-ldj}}$) results in a higher accuracy across tasks compared to standard adversarial training. Indeed, invertible network plays the role of "inductive bias" to alleviate the unavailability of labeled samples in the target domain. The term "inductive bias" refers to a set of assumptions that enhances the generalization ability of a model trained on empirical data distribution. As an example, a specific neural network architecture or a well-defined regularization can be regarded as inductive biases. Furthermore, when the loss including the term for the log determinant of Jacobian is applied (RIDA), our method demonstrates a significant improvement over $\text{RIDA}_{\text{no-ldj}}$ and previous works. These results demonstrate that log-determinant of Jacobian does make the model more robust to distortion caused by invariance enforcement.

Source **Target**	USPS MNIST	MNIST SVHN	SVHN MNIST	CIFAR10 STL10	STL10 CIFAR10
$\text{VADA}_{\text{no-vat}}$ (Shu et al., 2018)	-	66.8	83.1	79.1	68.6
$\text{RIDA}_{\text{no-ldj}}$	96.2	**73.3**	**92.5**	**80.2**	**71.4**
DIRT-T (Shu et al., 2018)	-	76.5	98.7	-	73.3
RIDA	**99.1**	**81.3**	**99.4**	**82.8**	**75.9**

TABLE 4.4. Test accuracy (%) on standard domain adaptation benchmarks in ablation experiment. The "no-ldj" subscript denotes models where the log-determinant of Jacobian loss is removed.



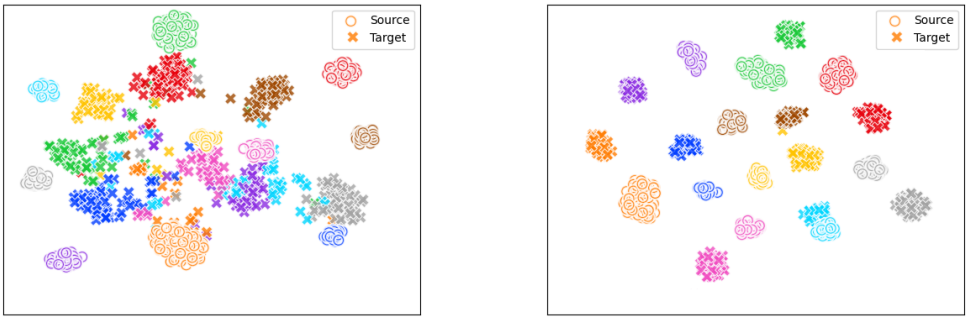
(a) MNIST → SVHN (b) SVHN → MNIST

FIGURE 4.5. Comparing the behavior of RIDA model with and without the log-determinant of Jacobian using the accuracy on the target domain for a) MNIST → SVHN, and b) SVHN → MNIST

4.5.5 Analysis

Qualitative Analysis

To further analyse the relaxed invariant representation, in Figure 4.6, we visualize the non-adapted and adapted feature representations generated from the last hidden layer of the model on SVHN → MNIST UDA task using t-SNE (Maaten and Hinton, 2008). As illustrated in Figure 4.6, source-only training or Non-adapted model shows strong clustering of the SVHN samples and performs poorly on MNIST (Figure 4.6a). RIDA delivers higher feature discriminability in the target domain by keeping each class well separated without enforcing the target clusters to be completely aligned with source domain (Figure 4.6b).

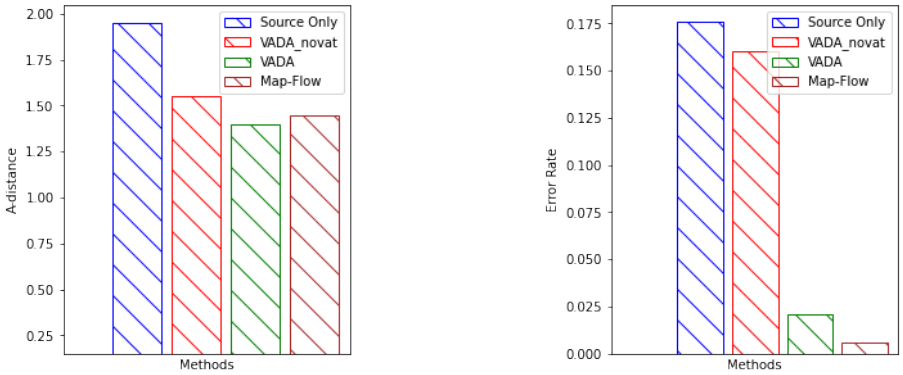


(a) Non-adapted (b) Adapted

FIGURE 4.6. t-SNE visualization of the last hidden layer of RIDA for SVHN → MNIST task a) Non-adapted, b) Adapted

Target Error Bound

We analyze the second term, domain discrepancy, and the third term, ideal joint hypothesis error, of the target error bound, as formulated in Equation 4.2, on SVHN → MNIST task.



(a) $\mathcal{A}$-distance (b) Ψ

FIGURE 4.7. $\mathcal{A}$-distance, and Ψ, evaluated on SVHN → MNIST task.

Domain Discrepancy. Domain discrepancy can be estimated approximately by $\mathcal{A}$-distance (Ben-David et al., 2010), defined as $\mathcal{A} = 2(1 - 2\epsilon)$, where ϵ denotes the error of a domain classifier trained to discriminate the source and target representations. As illustrated in Figure 4.7a, RIDA minimizes domain discrepancy more significantly than standard domain adversarial training, but does not as much as VADA (Shu et al., 2018) method does, implying a relaxed invariance.

Ideal Joint Hypothesis. We evaluate the Ideal Joint Hypothesis by training a MLP classifier with two layers on the adapted features from both target and source domains, as suggested in (Chen et al., 2019c). As shown in Figure 4.7b, RIDA reduces the joint error, which indicates that our method improves the feature discriminability.

4.6 Chapter Summary

In this chapter, a novel relaxed invariant representation learning is presented for unsupervised domain adaptation. In standard domain invariance learning, the transferability of feature representations is enhanced at the expense of its discriminability. Our method aims at preserving the discriminability by relaxing excessive invariance constraint. This is achieved by an invertible function that maps invariant features to a space that is easily separable for the classifier. Through extensive experiments, our approach demonstrates its superiority to other methods based on invariant representations on several public UDA datasets, validating our analysis.

Chapter 5

Conclusion and Future Work

The present thesis focused on unsupervised domain adaptation (UDA) that aims to transfer the knowledge extracted from label-abundant source domain to the unlabeled target domain. The difficulty is that it is not clear what specific factors of the data distribution remain invariant or change across domains. A representation of data that is invariant to domain changes is useful for reducing potential biases in prediction problems and controlling the effects of covariate shifts. Hence, learning invariant representations has recently attracted much attention for reconciling a source and a target domain for UDA. However, the domain-invariance learning may deteriorate the feature adaptability, though it improves the transferability of features. To resolve this problem, in this thesis, we investigated the limitation of invariant representation learning for unsupervised domain adaptation and proposed methods to relax the excessive invariance. In this chapter, we conclude the thesis by summarizing the main contributions and indicating possible directions for future research.

5.1 Contributions

We firstly provided a comprehensive literature review and required background on UDA algorithms related to image classification problems in Chapter 2. We then presented new approaches to address the excessive invariance enforcement.

In Chapter 3, we provided a theoretical discussion about the necessity of relaxed invariant representation, and then proposed a general relaxed-invariance framework for UDA. The framework relies on normalizing flow to learn a transformation between the distribution of target and source domains in representation space. In fact, normalizing flow maps a complex target latent distribution into a well-clustered latent source distribution through a sequence of invertible functions. We mathematically derived a variational lower bound for the probability distribution changing across domains and showed the consistency of the

lower bound with the relaxed invariance assumption. Although not empirically evaluated, this lower bound provided us with a new insight into UDA.

In Chapter 4, we developed a novel relaxed-invariant representation learning for unsupervised domain adaptation. This is achieved by adding an invertible network between representation space and embedding classifier, and then minimizing the log-determinant of the Jacobian of this network. Invertible networks ensure that the learnt features are not degenerate, and geometrically reshapes the invariant representation by scaling the features. Intuitively, weighting the embedding space via the determinant of Jacobian renders the features more robust to distortions caused by adversarial invariant learning. The extensive evaluation of the proposed method on benchmark datasets has validated the significance of the research direction by developing a simple methodology that outperforms state-of-the-art framework.

5.2 Future Work

Several possible avenues of future work are outlined below:

- Arguably, invariant representation is the key subject for research in domain adaptation. Thus, it is crucial to identify its open question for future investigation. In our proposed method, we have shown how adaptability in UDA is improved when invariant representation is relaxed. However, it is unclear what the general form of the adaptability-invariance trade-off is. Hence, the task of computing Pareto set, a set of Pareto optimal solutions, which expresses the trade-off between the conflicting objectives of adaptability and invariance, arises. As future work, we intend to develop Pareto-optimal algorithms for UDA by resorting to information-theoretic inference procedures and the theory of invariant representation learning.

- Learning representations that exhibit invariance across domains is a challenging task. Existing top-performance methods cast the trade-off between invariance and task performance in an adversarial way. In standard domain adversarial training (DAT), the discriminator (domain classifier), classifies representations as either source (positive) or target representation (negative). This binary (positive-negative) classification was kept fixed during the training process of the discriminator, without considering the fact that a representation of target might be extracted that may not be distinguishable from source representation at times. Thus, it is

better to treat the target representation as unlabeled, which can be either positive or negative. Hence, as future work, we will study how the positive-unlabelled learning role of discriminator affects the final prediction performance.

- In this thesis, we considered a close-set setting for UDA, where the source and the target distribution share the same class labels. However, if the label space changes across domains, the method may experience performance drop due to negative transfer learning, where transferring the knowledge from the labelled source domain negatively affects the target domain learner. Therefore, the proposed approach can be extended to the open set or partial UDA problems, settings where the source and target domains only share a subset of class labels, but not all class labels.

- The proposed domain adaptation method is evaluated in this thesis for the image classification task. For future work, we will extend our model to classification tasks that involve non-image data, and other domain-adaptation tasks such as semantic segmentation.

Appendices

A.1 Derivation of the ELBO

$$
\begin{aligned}
\log p(\mathbf{x}_t, \mathbf{x}_s) &= \log \int p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) p(\mathbf{z}_t, \mathbf{z}_s) d\mathbf{z}_s d\mathbf{z}_t \\
&= \log \int p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) p(\mathbf{z}_t, \mathbf{z}_s) \frac{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)}{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} d\mathbf{z}_s d\mathbf{z}_t \\
&= \log \mathbb{E}_{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} \big[p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) \frac{p(\mathbf{z}_t, \mathbf{z}_s)}{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} \big] \\
&\geq \mathbb{E}_{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} \big[\log \Big(p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) \frac{p(\mathbf{z}_t, \mathbf{z}_s)}{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} \Big) \big] \\
&= \mathbb{E}_{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} \big[\log(p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s)) + \log(p(\mathbf{z}_t, \mathbf{z}_s)) \big] \\
&\qquad - \mathbb{E}_{q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)} \big[\log(q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s)) \big]
\end{aligned}
\tag{A.1}
$$

The last result from the Equation A.1 is the ELBO of the the joint distribution. We further assume the conditional independence between source and target distribution:

$$
\begin{aligned}
q(\mathbf{z}_t, \mathbf{z}_s|\mathbf{x}_t, \mathbf{x}_s) &= q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s), \\
p(\mathbf{x}_t, \mathbf{x}_s|\mathbf{z}_t, \mathbf{z}_s) &= p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_s) p(\mathbf{x}_s|\mathbf{z}_s).
\end{aligned}
\tag{A.2}
$$

Therefore, we can derive the following result.

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} \big[\log(p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_s) p(\mathbf{x}_s|\mathbf{z}_s)) + \log(p(\mathbf{z}_t, \mathbf{z}_s)) - \log(q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)) \big] \\
&= \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} \big[\log(p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_s) + \log(p(\mathbf{x}_s|\mathbf{z}_s)) + \log(p(\mathbf{z}_t, \mathbf{z}_s)) \big] \\
&\qquad - \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} \big[\log(q(\mathbf{z}_t|\mathbf{x}_t)) - \log(q(\mathbf{z}_s|\mathbf{x}_s)) \big] \\
&= \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} \big[\log(p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_s) \big] + \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)} \big[\log(p(\mathbf{x}_s|\mathbf{z}_s)) \big] \\
&\qquad + \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} \big[\log(p(\mathbf{z}_t, \mathbf{z}_s)) - \log(q(\mathbf{z}_t|\mathbf{x}_t)) - \log(q(\mathbf{z}_s|\mathbf{x}_s)) \big] \\
&= \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} [\log(p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{z}_s)] + \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)} [\log(p(\mathbf{x}_s|\mathbf{z}_s))] \\
&\qquad + \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t) q(\mathbf{z}_s|\mathbf{x}_s)} [\log(p(\mathbf{z}_t, \mathbf{z}_s))] - \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)} [\log(q(\mathbf{z}_s|\mathbf{x}_s))] \\
&\qquad - \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)} [\log(q(\mathbf{z}_t|\mathbf{x}_t))].
\end{aligned}
\tag{A.3}
$$

We rewrite the last result of Equation A.3.

$$
\begin{aligned}
&= \underbrace{\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(p(\mathbf{x}_t|\mathbf{z}_t,\mathbf{z}_s)\big]}_{(1)} + \underbrace{\mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(p(\mathbf{x}_s|\mathbf{z}_s))\big]}_{(2)} \\
&+ \underbrace{\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(p(\mathbf{z}_t,\mathbf{z}_s))\big]}_{(3)} - \underbrace{\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)}\big[\log(q(\mathbf{z}_t|\mathbf{x}_t)\big]}_{(4)} \\
&\quad - \underbrace{\mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(q(\mathbf{z}_s|\mathbf{x}_s))\big]}_{(5)},
\end{aligned}
\tag{A.4}
$$

Nothing that

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{z}_t,\mathbf{z}_s) &= \frac{p(\mathbf{z}_t,\mathbf{z}_s|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_t,\mathbf{z}_s)} && \text{(Bayes' theorem)} \\
&= \frac{p(\mathbf{z}_t|\mathbf{z}_s,\mathbf{x}_t)p(\mathbf{z}_s|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_t,\mathbf{z}_s)} && \text{(Chain rule)} \\
&= \frac{p(\mathbf{z}_t|\mathbf{z}_s)p(\mathbf{z}_s|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_t,\mathbf{z}_s)} \\
&= \frac{p(\mathbf{z}_t|\mathbf{z}_s)p(\mathbf{z}_s|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_t|\mathbf{z}_s)p(\mathbf{z}_s)} \\
&= \frac{p(\mathbf{z}_s|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_s)} = \frac{p(\mathbf{z}_s,\mathbf{x}_t)}{p(\mathbf{z}_s)} = \frac{p(\mathbf{x}_t|\mathbf{z}_s)p(\mathbf{z}_s)}{p(\mathbf{z}_s)} = p(\mathbf{x}_t|\mathbf{z}_s),
\end{aligned}
\tag{A.5}
$$

then, the term (1) in Equation A.4 can be redefined as

$$
\begin{aligned}
(1) &= \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(p(\mathbf{x}_t|\mathbf{z}_t,\mathbf{z}_s)\big] = \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(p(\mathbf{x}_t|\mathbf{z}_s))\big] \\
&= \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(p(\mathbf{x}_t|\mathbf{z}_s))\big] = \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}\big[\log(\int p(\mathbf{x}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_s)d\mathbf{z}_t)\big] \\
&\geq \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}\big[\mathbb{E}_{p(\mathbf{z}_t|\mathbf{z}_s)}[\log(p(\mathbf{x}_t|\mathbf{z}_t))]\big].
\end{aligned}
\tag{A.6}
$$

By assuming that

$$
\mathbf{z}_t = f(\mathbf{z}_s) \quad \Rightarrow \quad p(\mathbf{z}_t|\mathbf{z}_s) = \delta(\mathbf{z}_t - f(\mathbf{z}_s)),
\tag{A.7}
$$

then, the term (3) in Equation A.4 can be redefined as

$$
\begin{aligned}
(3) &= \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}[\log(p(\mathbf{z}_t|\mathbf{z}_s)p(\mathbf{z}_s))] \\
&= \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}[\log(p(\mathbf{z}_t|\mathbf{z}_s))] + \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}[\log(p(\mathbf{z}_s))] \\
&= \text{constant} + \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)q(\mathbf{z}_s|\mathbf{x}_s)}[\log(p(\mathbf{z}_s)].
\end{aligned}
\tag{A.8}
$$

we also have

$$-\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)}\big[\log(q(\mathbf{z}_t|\mathbf{x}_t)] = -\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)}\big[\log(p(f^{-1}(\mathbf{z}_t)) + \log\Big|\det\frac{\partial f^{-1}}{\partial \mathbf{z}_t}\Big|\big], \quad \text{(A.9)}$$

Putting all them together we have the following final loss:

$$\begin{aligned}
\log(p(\mathbf{x}_t, \mathbf{x}_s)) &\geq \mathcal{L}(\boldsymbol{\theta}), \\
\text{where} \quad \mathcal{L}(\boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}[\mathbb{E}_{p(\mathbf{z}_t|\mathbf{z}_s)}[\log(p(\mathbf{x}_t|\mathbf{z}_t))]] + \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}[\log(p(\mathbf{x}_s|\mathbf{z}_s))] \\
&\quad + \mathbb{E}_{q(\mathbf{z}_s|\mathbf{x}_s)}[\log(p(\mathbf{z}_s)) - \log(q(\mathbf{z}_s|\mathbf{x}_s))] \\
&\quad - \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_t)}[\log(p(f^{-1}(\mathbf{z}_t)) + \log\Big|\det\frac{\partial f^{-1}}{\partial \mathbf{z}_t}\Big|]
\end{aligned}$$

A.2 Network Architectures

In this section, we provide the details of the network architecture of the Figure 4.1. For the adaptation tasks presented in Table 4.1, the shared classifier and encoder are adopted from DIRT-T (Shu et al., 2018). Table A.1 summarizes the RIDA architecture. We implement all the algorithms using PyTorch (Paszke et al., 2017).

Small	Large
$32 \times 32 \times 3$ Image	
3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU	3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU
2×2 max-pool, stride 2 dropout, p=0.5 Gaussian noise, $\sigma = 1$	
3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU	3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU 3×3 conv.64, BN, IReLU
2×2 max-pool, stride 2 dropout, p=0.5 Gaussian noise, $\sigma = 1$	
4 affine coupling blocks 3 FC Layer 256 unit	4 affine coupling blocks 6 FC Layer 512 unit
2×2 max-pool, stride 2 dropout, p=0.5 Gaussian noise, $\sigma = 1$	
3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU	3×3 conv.64, BN, lReLU 3×3 conv.64, BN, lReLU 3×3 conv.64, BN, IReLU
global average pool	
10 dense, softmax	

TABLE A.1. Small and large network archhitectures for different adaptation tasks. Leaky ReLU parameter $\alpha = 0.1$. All images are resized to $32 \times 32 \times 3$.

For office-31 and VisDA 2017 datasets, we employ ResNet-50 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) as the feature extractor.

The discriminator network consists of three fully connected layers with dropout as other works (Ganin et al., 2016). The invertible network architecture is the same as invertible layers of large architectures in Table 4.1.

A.3 Hyper-parameters

Table A.2 presents the hyper-parameters used in our experiments. We followed a similar hyper-parameter selection strategy as in DIRT-T (Shu et al., 2018), for digits and CIFAR10/STL datasets, and (Chen et al., 2020) for office-31 and VisDA datasets. Hyper-parameter values are searched within the following predefined space: $\beta_d = \{0, 1, 10^{-2}\}, \beta_t = \{1, 10\}, \beta_s = \{0, 1\}$, and $\beta_{ce} = \{0, 10^{-1}, 10^{-2}\}$.

Experiment	β_d	β_t	β_s	β_{ce}
USPS $\rightarrow$ MNIST	10^{-2}	1	0	10^{-2}
MNIST $\rightarrow$ SVHN	10^{-2}	1	1	10^{-2}
SVHN $\rightarrow$ MNIST	10^{-2}	1	0	10^{-2}
CIFAR10 $\rightarrow$ STL	10^{-2}	1	0	10^{-1}
STL $\rightarrow$ CIFAR10	0	1	1	10^{-1}
A $\rightarrow$ W	1	1	0	10^{-1}
A $\rightarrow$ D	1	1	1	10^{-1}
W $\rightarrow$ A	1	1	1	10^{-1}
VisDA-2017 Classification	1	10	0	10^{-2}

TABLE A.2. Hyper-parameters for the tasks in the experiments with SOTA results.

Bibliography

Adler, Jonas and Sebastian Lunz (2018). "Banach wasserstein gan". In: *arXiv preprint arXiv:1806.06621*.

Arandjelovic, Relja, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic (2016). "NetVLAD: CNN architecture for weakly supervised place recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307.

Arazo, Eric, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness (2020). "Pseudo-labeling and confirmation bias in deep semi-supervised learning". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8.

Arjovsky (2017). "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR, pp. 214–223.

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz (2019). "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893*.

Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan (2010). "A theory of learning from different domains". In: *Machine learning* 79.1-2, pp. 151–175.

Ben-David, Shai, John Blitzer, Koby Crammer, and Fernando Pereira (2007). "Analysis of representations for domain adaptation". In: *Advances in neural information processing systems*, pp. 137–144.

Blum, Avrim and Tom Mitchell (1998). "Combining labeled and unlabeled data with co-training". In: *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100.

Bonneel, Nicolas, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister (2015). "Sliced and radon wasserstein barycenters of measures". In: *Journal of Mathematical Imaging and Vision* 51.1, pp. 22–45.

Bouvier, Victor, Céline Hudelot, Clément Chastagnol, Philippe Very, and Myriam Tami (2019a). "Domain-Invariant Representations: A Look on Compression and Weights". In:

Bouvier, Victor, Philippe Very, Céline Hudelot, and Clément Chastagnol (2019b). "Hidden covariate shift: A minimal assumption for domain adaptation". In: *arXiv preprint arXiv:1907.12299*.

Cao, Yue, Mingsheng Long, and Jianmin Wang (2018). "Unsupervised domain adaptation with distribution matching machines". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Cao, Zhangjie, Lijia Ma, Mingsheng Long, and Jianmin Wang (2018). "Partial adversarial domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 135–150.

Carlucci, Fabio Maria, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo (2017). "Autodial: Automatic domain alignment layers". In: *2017 IEEE international conference on computer vision (ICCV)*. IEEE, pp. 5077–5085.

Chapelle, Olivier and Alexander Zien (2005). "Semi-supervised classification by low density separation." In: *AISTATS*. Vol. 2005. Citeseer, pp. 57–64.

Chen, Chaoqi, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang (2019a). "Progressive feature alignment for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 627–636.

Chen, Dong-Dong, Yisen Wang, Jinfeng Yi, Zaiyi Chen, and Zhi-Hua Zhou (2019b). "Joint Semantic Domain Alignment and Target Classifier Learning for Unsupervised Domain Adaptation". In: *arXiv preprint arXiv:1906.04053*.

Chen, Minghao, Shuai Zhao, Haifeng Liu, and Deng Cai (2020). "Adversarial-learned loss for domain adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3521–3528.

Chen, Qingchao, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty (2018). "Re-weighted adversarial adaptation network for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7976–7985.

Chen, Xinyang, Sinan Wang, Mingsheng Long, and Jianmin Wang (2019c). "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation". In: *International conference on machine learning*. PMLR, pp. 1081–1090.

Chopra, Sumit, Suhrid Balakrishnan, and Raghuraman Gopalan (2013). "Dlid: Deep learning for domain adaptation by interpolating between domains". In: *ICML workshop on challenges in representation learning*. Vol. 2. 6. Citeseer.

Cicek, Safa and Stefano Soatto (2019). "Unsupervised domain adaptation via regularized conditional alignment". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1416–1425.

Coates, Adam, Andrew Ng, and Honglak Lee (2011). "An analysis of single-layer networks in unsupervised feature learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223.

Coates, Adam and Andrew Y Ng (2012). "Learning feature representations with k-means". In: *Neural networks: Tricks of the trade*. Springer, pp. 561–580.

Combes, Remi Tachet des, Han Zhao, Yu-Xiang Wang, and Geoff Gordon (2020). "Domain adaptation with conditional distribution matching and generalized label shift". In: *arXiv preprint arXiv:2003.04475*.

Courty, Nicolas, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy (2017). "Joint distribution optimal transportation for domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 3730–3739.

Cui, Shuhao, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian (2020a). "Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3941–3950.

Cui, Shuhao, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian (2020b). "Gradually vanishing bridge for adversarial domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12455–12464.

Damodaran, Bharath Bhushan, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty (2018). "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation". In: *European Conference on Computer Vision*. Springer, pp. 467–483.

Deng, Zhijie, Yucen Luo, and Jun Zhu (2019). "Cluster alignment with a teacher for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9944–9953.

Dinh, Laurent, David Krueger, and Yoshua Bengio (2014). "Nice: Non-linear independent components estimation". In: *arXiv preprint arXiv:1410.8516*.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). "Density estimation using real nvp". In: *arXiv preprint arXiv:1605.08803*.

Donahue, Jeff, Philipp Krähenbühl, and Trevor Darrell (2016). "Adversarial feature learning". In: *arXiv preprint arXiv:1605.09782*.

Dumoulin, Vincent, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville (2016). "Adversarially learned inference". In: *arXiv preprint arXiv:1606.00704*.

Elhadji-Ille-Gado, Nassara, Edith Grall-Maes, and Malika Kharouf (2017). "Transfer learning for large scale data using subspace alignment". In: *2017 16th*

IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp. 1006–1010.

Fernando, Basura, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars (2013). "Unsupervised visual domain adaptation using subspace alignment". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2960–2967.

French (2017). "Self-ensembling for visual domain adaptation". In: *arXiv preprint arXiv:1706.05208*.

Ganin, Yaroslav and Victor Lempitsky (2014). "Unsupervised domain adaptation by backpropagation". In: *arXiv preprint arXiv:1409.7495*.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky (2016). "Domain-adversarial training of neural networks". In: *The Journal of Machine Learning Research* 17.1, pp. 2096–2030.

Ghifary, Muhammad, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li (2016). "Deep reconstruction-classification networks for unsupervised domain adaptation". In: *European Conference on Computer Vision*. Springer, pp. 597–613.

Gong, Boqing, Yuan Shi, Fei Sha, and Kristen Grauman (2012). "Geodesic flow kernel for unsupervised domain adaptation". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2066–2073.

Gong, Rui, Wen Li, Yuhua Chen, and Luc Van Gool (2019). "Dlow: Domain flow for adaptation and generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2477–2486.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). *Deep learning*. Vol. 1. 2. MIT press Cambridge.

Goodfellow, Ian J, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial networks". In: *arXiv preprint arXiv:1406.2661*.

Gopalan, Raghuraman, Ruonan Li, and Rama Chellappa (2011). "Domain adaptation for object recognition: An unsupervised approach". In: *2011 international conference on computer vision*. IEEE, pp. 999–1006.

Grandvalet, Yves, Yoshua Bengio, et al. (2005). "Semi-supervised learning by entropy minimization." In: *CAP*, pp. 281–296.

Grover, Aditya, Christopher Chute, Rui Shu, Zhangjie Cao, and Stefano Ermon (2019). "AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows". In: *arXiv preprint arXiv:1905.12892*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hoffman, Judy, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell (2017). "Cycada: Cycle-consistent adversarial domain adaptation". In: *arXiv preprint arXiv:1711.03213*.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR, pp. 448–456.

Iuchi, Hitoshi, Taro Matsutani, Keisuke Yamada, Natsuki Iwano, Shunsuke Sumi, Shion Hosoda, Shitao Zhao, Tsukasa Fukunaga, and Michiaki Hamada (2021). "Representation learning applications in biological sequence analysis". In: *bioRxiv*.

Jiang, Xiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei (2020). "Implicit class-conditioned domain alignment for unsupervised domain adaptation". In: *International Conference on Machine Learning*. PMLR, pp. 4816–4827.

Jin, Ying, Ximei Wang, Mingsheng Long, and Jianmin Wang (2020). "Minimum Class Confusion for Versatile Domain Adaptation". In: *European Conference on Computer Vision*. Springer, pp. 464–480.

Johansson, Fredrik D, Rajesh Ranganath, and David Sontag (2019). "Support and invertibility in domain-invariant representations". In: *arXiv preprint arXiv:1903.03448*.

Kang, Guoliang, Lu Jiang, Yi Yang, and Alexander G Hauptmann (2019). "Contrastive Adaptation Network for Unsupervised Domain Adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902.

Kim, Taeksoo, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim (2017). "Learning to discover cross-domain relations with generative adversarial networks". In: *International Conference on Machine Learning*. PMLR, pp. 1857–1865.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kingma, Durk P and Prafulla Dhariwal (2018). "Glow: Generative flow with invertible 1x1 convolutions". In: *Advances in Neural Information Processing Systems*, pp. 10215–10224.

Krizhevsky, Alex, Geoffrey Hinton, et al. (2009). *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25, pp. 1097–1105.

Kumar, Abhishek, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell (2018). "Co-regularized alignment for unsupervised domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 9345–9356.

Laine, Samuli and Timo Aila (2016). "Temporal ensembling for semi-supervised learning". In: *arXiv preprint arXiv:1610.02242*.

Le, Trung, Khanh Nguyen, Nhat Ho, Hung Bui, and Dinh Phung (2018). "On Deep Domain Adaptation: Some Theoretical Understandings". In: *arXiv preprint arXiv:1811.06199*.

Le Cun, Yann, Ofer Matan, Bernhard Boser, John S Denker, Don Henderson, Richard E Howard, Wayne Hubbard, LD Jackel, and Henry S Baird (1990). "Handwritten zip code recognition with multilayer networks". In: *Proc. 10th International Conference on Pattern Recognition*. Vol. 2, pp. 35–40.

LeCun, Yann (1998). "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/*.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

Lee, Chen-Yu, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht (2019a). "Sliced wasserstein discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10285–10295.

Lee, Seungmin, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong (2019b). "Drop to adapt: Learning discriminative features for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 91–100.

Li, Chunyuan, Hao Liu, Changyou Chen, Yunchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin (2017). "Alice: Towards understanding adversarial learning for joint distribution matching". In: *arXiv preprint arXiv:1709.01215*.

Li, Yanghao, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou (2016). "Revisiting batch normalization for practical domain adaptation". In: *arXiv preprint arXiv:1603.04779*.

Liang, Jian, Ran He, Zhenan Sun, and Tieniu Tan (2018). "Aggregating randomized clustering-promoting invariant projections for domain adaptation".

In: *IEEE transactions on pattern analysis and machine intelligence* 41.5, pp. 1027–1042.

Lim, Jae Hyun and Jong Chul Ye (2017). "Geometric gan". In: *arXiv preprint arXiv:1705.02894*.

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

Liu, Feng, Jie Lu, Bo Han, Gang Niu, Guangquan Zhang, and Masashi Sugiyama (2019a). "Butterfly: A panacea for all difficulties in wildly unsupervised domain adaptation". In: *arXiv preprint arXiv:1905.07720*.

Liu, Hong, Mingsheng Long, Jianmin Wang, and Michael I Jordan (2019b). "Transfer Adversarial Training: A General Approach to Adapting Deep Classifiers". In: *Transfer* 1/20.

Liu, Ming-Yu, Thomas Breuel, and Jan Kautz (2017). "Unsupervised image-to-image translation networks". In: *arXiv preprint arXiv:1703.00848*.

Liu, Ming-Yu and Oncel Tuzel (2016). "Coupled generative adversarial networks". In: *Advances in neural information processing systems*, pp. 469–477.

Long, Mingsheng, Yue Cao, Jianmin Wang, and Michael I Jordan (2015). "Learning transferable features with deep adaptation networks". In: *arXiv preprint arXiv:1502.02791*.

Long, Mingsheng, Zhangjie Cao, Jianmin Wang, and Michael I Jordan (2018). "Conditional adversarial domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 1640–1650.

Long, Mingsheng, Han Zhu, Jianmin Wang, and Michael I Jordan (2016). "Unsupervised domain adaptation with residual transfer networks". In: *Advances in Neural Information Processing Systems*, pp. 136–144.

Long, Mingsheng, Han Zhu, Jianmin Wang, and Michael I Jordan (2017). "Deep transfer learning with joint adaptation networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2208–2217.

Maaten, Laurens Van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11.

Magliacane, Sara, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij (2017). "Domain adaptation by using causal inference to predict invariant conditional distributions". In: *arXiv preprint arXiv:1707.06422*.

Miyato, Takeru, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii (2018). "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8, pp. 1979–1993.

Morerio, Pietro, Jacopo Cavazza, and Vittorio Murino (2017). "Minimal-entropy correlation alignment for unsupervised deep domain adaptation". In: *arXiv preprint arXiv:1711.10288*.

Motiian, Saeid, Quinn Jones, Seyed Mehdi Iranmanesh, and Gianfranco Doretto (2017). "Few-shot adversarial domain adaptation". In: *arXiv preprint arXiv:1711.02536*.

Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng (2011). "Reading digits in natural images with unsupervised feature learning". In:

Park, Sungrae, JunKeon Park, Su-Jin Shin, and Il-Chul Moon (2018). "Adversarial dropout for supervised and semi-supervised learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer (2017). "Automatic differentiation in pytorch". In:

Pei, Zhongyi, Zhangjie Cao, Mingsheng Long, and Jianmin Wang (2018). "Multi-adversarial domain adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Peng, Kuan-Chuan, Ziyan Wu, and Jan Ernst (2018). "Zero-shot deep domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 764–781.

Peng, Xingchao, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko (2017). "Visda: The visual domain adaptation challenge". In: *arXiv preprint arXiv:1710.06924*.

Peng, Xue Bin, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine (2018). "Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow". In: *arXiv preprint arXiv:1810.00821*.

Pérez-Carrasco, Manuel, Guillermo Cabrera-Vives, Pavlos Protopapas, Nicolás Astorga, and Marouan Belhaj (2019). "Matching Embeddings for Domain Adaptation". In: *arXiv preprint arXiv:1909.11651*.

Prabhu, Viraj, Shivam Khare, Deeksha Kartik, and Judy Hoffman (2020). "SENTRY: Selective Entropy Optimization via Committee Consistency for Unsupervised Domain Adaptation". In: *arXiv preprint arXiv:2012.11460*.

Radford, Alec, Luke Metz, and Soumith Chintala (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434*.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250*.

Rezende, Danilo and Shakir Mohamed (2015). "Variational inference with normalizing flows". In: *International Conference on Machine Learning*. PMLR, pp. 1530–1538.

Roy, Subhankar, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci (2019). "Unsupervised domain adaptation using feature-whitening and consensus loss". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9471–9480.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252.

Saenko, Kate, Brian Kulis, Mario Fritz, and Trevor Darrell (2010). "Adapting visual category models to new domains". In: *European conference on computer vision*. Springer, pp. 213–226.

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

Saito, Kuniaki, Yoshitaka Ushiku, and Tatsuya Harada (2017). "Asymmetric tri-training for unsupervised domain adaptation". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2988–2997.

Saito, Kuniaki, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko (2017). "Adversarial dropout regularization". In: *arXiv preprint arXiv:1711.01575*.

Saito, Kuniaki, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada (2018). "Maximum classifier discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732.

Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen (2016). "Improved techniques for training gans". In: *arXiv preprint arXiv:1606.03498*.

Sener, Ozan, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese (2016). "Learning transferrable representations for unsupervised domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 2110–2118.

Shen, Jian, Yanru Qu, Weinan Zhang, and Yong Yu (2017). "Wasserstein distance guided representation learning for domain adaptation". In: *arXiv preprint arXiv:1707.01217*.

Shimodaira, Hidetoshi (2000). "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Shu, Rui, Hung H Bui, Hirokazu Narui, and Stefano Ermon (2018). "A dirt-t approach to unsupervised domain adaptation". In: *arXiv preprint arXiv:1802.08735*.

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. (2017). "Mastering the game of go without human knowledge". In: *nature* 550.7676, pp. 354–359.

Sun, Baochen and Kate Saenko (2016). "Deep coral: Correlation alignment for deep domain adaptation". In: *European Conference on Computer Vision*. Springer, pp. 443–450.

Sun, Yu, Eric Tzeng, Trevor Darrell, and Alexei A Efros (2019). "Unsupervised domain adaptation through self-supervision". In: *arXiv preprint arXiv:1909.11825*.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tarvainen, Antti and Harri Valpola (2017). "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *arXiv preprint arXiv:1703.01780*.

Thopalli, Kowshik, Rushil Anirudh, Jayaraman J Thiagarajan, and Pavan Turaga (2019). "Multiple subspace alignment improves domain adaptation". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3552–3556.

Tzeng, Eric, Judy Hoffman, Trevor Darrell, and Kate Saenko (2015). "Simultaneous deep transfer across domains and tasks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 4068–4076.

Tzeng, Eric, Judy Hoffman, Kate Saenko, and Trevor Darrell (2017). "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176.

Tzeng, Eric, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell (2014). "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474*.

Vincent, Pascal, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou (2010). "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." In: *Journal of machine learning research* 11.12.

Volpi, Riccardo, Pietro Morerio, Silvio Savarese, and Vittorio Murino (2018). "Adversarial feature augmentation for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5495–5504.

Wang, Hao, Hao He, and Dina Katabi (2020). "Continuously Indexed Domain Adaptation". In: *arXiv preprint arXiv:2007.01807*.

Wang, Ximei, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael Jordan (2019a). "Transferable normalization: Towards improving transferability of deep neural networks". In:

Wang, Ximei, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang (2019b). "Transferable attention for domain adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 5345–5352.

Wen, Jun, Risheng Liu, Nenggan Zheng, Qian Zheng, Zhefeng Gong, and Junsong Yuan (2019). "Exploiting local feature patterns for unsupervised domain adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 5401–5408.

Wu, Yifan, Ezra Winston, Divyansh Kaushik, and Zachary Lipton (2019). "Domain Adaptation with Asymmetrically-Relaxed Distribution Alignment". In: *arXiv preprint arXiv:1903.01689*.

Xu, Ruijia, Guanbin Li, Jihan Yang, and Liang Lin (2019). "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1426–1435.

Yan, Hongliang, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo (2017). "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2272–2281.

You, Kaichao, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan (2019). "Universal domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2720–2729.

Yue, Xiangyu, Bichen Wu, Sanjit A Seshia, Kurt Keutzer, and Alberto L Sangiovanni-Vincentelli (2018). "A lidar point cloud generator: from a virtual world to autonomous driving". In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 458–464.

Zeng, Min, Min Li, Zhihui Fei, Ying Yu, Yi Pan, and Jianxin Wang (2019). "Automatic ICD-9 coding via deep transfer learning". In: *Neurocomputing* 324, pp. 43–50.

Zhang, Hongyi, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz (2017). "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412*.

Zhang, Jing, Wanqing Li, Philip Ogunbona, and Dong Xu (2019a). "Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective". In: *ACM Computing Surveys (CSUR)* 52.1, pp. 1–38.

Zhang, Weichen, Wanli Ouyang, Wen Li, and Dong Xu (2018a). "Collaborative and adversarial network for unsupervised domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3801–3809.

Zhang, Yabin, Hui Tang, Kui Jia, and Mingkui Tan (2019b). "Domain-symmetric networks for adversarial domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5031–5040.

Zhang, Yuchen, Tianle Liu, Mingsheng Long, and Michael Jordan (2019c). "Bridging theory and algorithm for domain adaptation". In: *International Conference on Machine Learning*. PMLR, pp. 7404–7413.

Zhang, Yun, Nianbin Wang, Shaobin Cai, and Lei Song (2018b). "Unsupervised domain adaptation by mapped correlation alignment". In: *IEEE Access* 6, pp. 44698–44706.

Zhao, Han, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon (2019). "On learning invariant representations for domain adaptation". In: *International Conference on Machine Learning*. PMLR, pp. 7523–7532.

Zhao, Junbo, Michael Mathieu, and Yann LeCun (2016). "Energy-based generative adversarial network". In: *arXiv preprint arXiv:1609.03126*.

Zhou, Zhi-Hua and Ming Li (2005). "Tri-training: Exploiting unlabeled data using three classifiers". In: *IEEE Transactions on knowledge and Data Engineering* 17.11, pp. 1529–1541.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A Efros (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Zhu, Lei, Wei Wang, Mei Hui Zhang, Beng Chin Ooi, and Chang Yao (2019). "Distribution Matching Prototypical Network for Unsupervised Domain Adaptation". In:

Zhur, Xiaojin and Zoubin Ghahramani (2002). "Learning from labeled and unlabeled data with label propagation". In:

Zou, Han, Yuxun Zhou, Jianfei Yang, Huihan Liu, Hari Prasanna Das, and Costas J Spanos (2019). "Consensus adversarial domain adaptation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 5997–6004.