Multi-Domain Text Classification with Adversarial Training

by

Yuan Wu

A thesis submitted to
the Faculty of Graduate and Postdoctoral Affairs
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science

Ottawa-Carleton Institute for Computer Science (OCICS)

School of Computer Science
Carleton University
Ottawa, Ontario, Canada

May 2022

Copyright ©
2022 - Yuan Wu

Abstract

Text classification is one of the fundamental tasks in natural language processing (NLP), which has been studied for decades and various approaches have been proposed. Unfortunately, text classification is a highly domain-dependent task, a subtle shift between training and testing data distributions can cause catastrophic performance deterioration. Moreover, the availability of massive labeled data varies among different domains in real-world applications. Therefore, it is of great importance to investigate how to improve the classification accuracy of the target domain by leveraging resources from related domains.

Multi-domain text classification (MDTC) is proposed to address the above problem. Nowadays, the mainstream MDTC approaches resort to transfer learning techniques to reduce domain divergence across different domains. In particular, these methods adopt adversarial training and shared-private paradigm to implement domain alignment, yielding state-of-the-art performance. Adversarial learning can reduce domain divergence through a minimax optimization to produce domain-invariant features. The domain-invariant features are supposed to be both transferable and discriminative, while shared-private employs domain-specific features to boost the discriminability of the domain-invariant features.

In this thesis, we make several contributions to advance MDTC: First, we propose a dual adversarial co-learning method that utilizes two forms of adversarial training to refine domain-variant features. Second, we apply mixup to conduct the category and

domain regularizations to enrich the intrinsic features in the shared latent space and enforce the consistent predictions in-between training samples such that the learned features can be more transferable and discriminative. Third, we analyze the limitation of the adversarial alignment on marginal distributions and propose a novel conditional adversarial network that aligns joint distributions of domain-invariant features and label predictions. Fourth, we propose a co-regularized adversarial learning framework that constructs two diverse adversarial training streams and aligns multiple conditional distributions by penalizing the disagreements of outputs of these two streams. We also incorporate entropy minimization and virtual adversarial training to avoid the violation of the cluster assumption. Finally, we incorporate the margin discrepancy to measure the domain divergence for MDTC and fill the gap between MDTC algorithms and theories by deriving a new generalization bound based on the margin discrepancy.

To my parents

Acknowledgments

A PhD is a long journey through unknown lands, over sky-high peaks, and through deep and dark thoughts. I am grateful to many people who accompanied me on this journey. First and foremost, I would like to thank my PhD supervisors Dr. Ahmed El-Roby and Dr. Diana Inkpen for their fundamental roles in my doctoral research. The freedom they gave me to pursue my interests and follow my curiosity has made all the difference. I appreciate their sensible advice, patience and endless support. They not only guide me in my research, but also teach me how to be a nice person.

I want to thank my beloved partner Lingling Li who supported me and encouraged me through the whole journey. I am also thankful to my parents for their limitless love and support.

I am very fortunate to work with Dr. Michiel Smid, Dr. Pat Morin, Dr. Mattew Holden, and Dr. Darryl Hill. They taught me how to communicate with undergraduate students and be responsible to a specific course. Working with them has been one of the enjoyable points of this PhD journey.

I would like to express my gratitude to all the staff members in the Ottawa-Carleton Institute for Computer Science, for their constant presence and optimistic atmosphere. I am grateful to my examiners Dr. Samira Sadoui, Dr. John Oommen, Dr. Paula Branco, Dr. Omair Shafiq, and Dr. Elinor Sloan for their constructive comments. I also thank my colleagues Yan Yan, Xuejun Han, and Hao Yan. Lastly, my special thank to Dr. Frank Dehne, for saving me from my lowest.

Preface

Publication included for credit

1. Wu, Y., & Guo, Y. (2020, April). Dual adversarial co-learning for multi-domain text classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 6438-6445).

2. Wu, Y., Inkpen, D., & El-Roby, A. (2020, August). Dual mixup regularized learning for adversarial domain adaptation. In European Conference on Computer Vision (pp. 540-555).

3. Wu, Y., Inkpen, D., & El-Roby, A. (2021). Mixup Regularized Adversarial Networks for Multi-Domain Text Classification. In proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

4. Wu, Y., Inkpen, D., & El-Roby, A. (2021). Conditional Adversarial Networks for Multi-Domain Text Classification. In the Second Workshop on Domain Adaptation for NLP at the 16th Conference of the European Chapter of the Association for Computational Linguistics.

5. Wu, Y., Inkpen, D., & El-Roby, A. (2021). Towards Category and Domain Alignment: Category-Invariant Feature Enhancement for Adversarial Domain Adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 132-141).

6. Wu, Y., Inkpen, D., & El-Roby, A. (2022). Maximum Batch Frobenius Norm for Multi-Domain Text Classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

7. Wu, Y., Inkpen, D., & El-Roby, A. (2022). Co-Regularized Adversarial Learning for Multi-Domain Text Classification. In International Conference on Artificial Intelligence and Statistics. PMLR.

Table of Contents

Abstract **ii**

Acknowledgments **v**

Preface **vi**

Table of Contents **viii**

List of Tables **xiii**

List of Figures **xv**

1 Introduction **1**

1.1 Overview 1

1.2 Research Problems 4

1.3 Contributions 5

1.4 Outline 6

1.5 Summary 8

2 Related Work **10**

2.1 Adversarial Learning 10

2.2 Domain Adaptation 11

2.2.1 Information Theoretical Measures 12

2.2.2 Higher-Order Moment Measures 13
2.2.3 Statistical-based Approaches 16
2.2.4 Adversarial Learning-based Approaches 17
2.3 Multi-task learning 19
2.4 Multi-domain text classification 21
2.4.1 Task formulation 23
2.4.2 Evaluation metrics 23
2.5 Summary 24

3 Dataset **25**

3.1 The Amazon Review Dataset 25
3.2 The FDU-MTL Dataset 26
3.3 Summary 27

4 Dual Adversarial Co-Learning for Multi-Domain Text Classification **29**

4.1 Introduction 29
4.2 Dual Adversarial Co-Learning 31
4.2.1 Domain Discrimination based Adversarial Learning 33
4.2.2 Separation regularizer 34
4.2.3 Prediction Discrepancy based Adversarial Learning 34
4.2.4 Stepwise Adversarial Training 36
4.3 Experiments 39
4.3.1 Implementation Details 39
4.3.2 Comparison baselines 39
4.3.3 Experimental Results 41
4.3.4 Multi-Source Unsupervised Domain Adaptation 42
4.3.5 Ablation Study 43

4.3.6 Parameter Sensitivity Analysis 44
4.4 Summary 45

5 Mixup Regularized Adversarial Networks for Multi-Domain Text Classification 48

5.1 Introduction 48
5.2 Mixup Regularized Adversarial Learning 50
5.2.1 Multinomial Adversarial Learning 51
5.2.2 Mixup-based Regularization 52
5.2.3 Training Procedure 54
5.3 Experiments 55
5.3.1 Comparison Methods 55
5.3.2 Implementation Details 56
5.3.3 Results 57
5.3.4 Multi-Source Unsupervised Domain Adaptation 58
5.3.5 Ablation Study 60
5.3.6 Parameter Sensitivity Analysis 60
5.4 Summary 61

6 Conditional Adversarial Networks for Multi-Domain Text Classification 65

6.1 Introduction 65
6.2 Conditional Adversarial Learning 67
6.2.1 Entropy Conditioning 70
6.2.2 Theoretical Analysis 71
6.3 Proof 72
6.4 Experiments 76
6.4.1 Implementation Details 76

6.4.2 Comparison Methods 76
6.4.3 Results 78
6.4.4 Multi-Source Unsupervised Domain Adaptation 79
6.4.5 Ablation Study 80
6.4.6 Parameter Sensitivity Analysis 80
6.5 Summary 81

7 Co-Regularized Adversarial Learning for Multi-Domain Text Classification 85

7.1 Introuction 85
7.2 Model Architecture 88
7.3 Multinomial Adversarial Learning 89
7.4 Co-Regularized Adversarial Learning 90
7.5 Virtual Adversarial Training 92
7.6 The Final Objective 93
7.7 Discussion 95
7.8 Experiments 96
7.8.1 Implementation Details 96
7.8.2 Comparison Methods 97
7.8.3 Results 99
7.8.4 Further Analysis 101
7.9 Summary 103

8 Margin Discrepancy-based Adversarial Training for Multi-Domain Text Classification 105

8.1 Introduction 105
8.2 Preliminaries 107
8.2.1 Basic Notation 107

8.2.2 Adversarial Multi-Domain Text Classification 108
8.2.3 Divergence Measurements 109
8.2.4 Margin Discrepancy 110
8.3 Methodology 111
8.3.1 Theoretical Analysis 112
8.3.2 Theoretical Proof 114
8.3.3 Margin Discrepancy-based Adversarial Training 120
8.3.4 Training procedure 122
8.4 Experiments 124
8.4.1 Implementation Details 124
8.4.2 Comparison Methods 126
8.4.3 Results 127
8.5 Further Analysis 128
8.5.1 Ablation Study 128
8.5.2 Parameter Sensitivity Analysis 129
8.5.3 Multi-Source Unsupervised Domain Adaptation 130
8.6 Summary 131

9 Conclusion and Future Work **133**
9.1 Conclusion 133
9.2 Future Work 134

List of References **136**

List of Tables

1 Statistics of the Amazon review dataset 26
2 Statistics of the FDU-MTL dataset 28
3 The classification accuracies of DACL on the Amazon review dataset. 41
4 The classification accuracies of DACL on the FDU-MTL dataset. . . 46
5 MS-UDA classification accuracies of DACL on the Amazon review dataset 47
6 Ablation study analysis of the DACL. 47
3 The classification accuracies of MRAN on the Amazon review dataset. 58
4 The classification accuracies of MRAN on the FDU-MTL dataset. . 63
5 Multi-source unsupervised domain adaptation results of MRAN on the Amazon review dataset. 64
6 Ablation study analysis of MRAN. 64
7 The classification accuracies of CAN on the Amazon review dataset. . 78
8 The classification accuracies of CAN on the FDU-MTL dataset. . . . 83
9 Multi-source unsupervised domain adaptation results of CAN on the Amazon review dataset. 84
10 Ablation study analysis of CAN. 84
11 The classification accuracies of CRAL on the Amazon review dataset. 99
12 The classification accuracies of CRAL on the FDU-MTL dataset. . . 100

13 The Multi-source unsupervised domain adaptation results of CRAL on the Amazon review dataset. 102
14 Ablation study of CRAL. 103
15 The classification accuracies of MDAT on the Amazon review dataset. 124
16 The classification accuracies of MDAT on the FDU-MTL dataset. . . 125
17 Ablation Study on the Amazon review dataset 129
18 Multi-source unsupervised domain adaptation results of MDAT. . . . 130

List of Figures

1 The training and testing processes of the DANN. 18

2 Two paradigms for tackling MDTC tasks. 23

3 Architecture of the dual adversarial co-learning model. 32

4 Parameter sensitivity analysis of the DACL 44

5 The architecture of the MRAN. 50

6 Parameter sensitivity analysis of MRAN 61

7 The architecture of the CAN model. 68

8 The parameter sensitivity analysis of CAN. 81

9 The Architecture of the CRAL framework. 88

10 Examples for the domain alignment in MDTC. 91

11 Parameter sensitivity analysis of CRAL 104

12 The architecture of the MDAT method. 114

13 Parameter sensitivity analysis of MDAT 129

Chapter 1

Introduction

1.1 Overview

In recent years, with the rapid development of the internet and social media, a huge amount of text data is flooding every field of our life. Most of the text data is in the form of unstructured raw data. The unstructured text data are easy to cause trouble for people to find what they exactly need. Developing systems that can understand unstructured text data and reasonably infer its sentiment is one of the main challenges for artificial intelligence. This objective drives the research in machine learning and natural language processing (NLP).

Text classification in that context is the process of assigning sentiment labels to text data according to its detailed content. It is one of the most fundamental tasks of natural language processing. In the past decades, researchers have developed many text classification algorithms from the statistical perspective [1]. These methods required manually generated features to perform text classification. The performance of these models is heavily reliant on the effectiveness of the manually engineering features. These models are limited to understanding the target text data as the manually engineered features are proven to be restrictive to tell a model what factors in the data are key to make its decisions [2]. Moreover, manually engineering features

are time-consuming, laborious, and of high cost.

Nowadays, the automatic feature extraction technique has emerged with the advent of deep neural networks (DNNs) [3]. It can process huge amounts of text data quickly and accurately. DNN-based text classification models have drawn more and more attention in recent years both in academia and industry. Many breakthroughs in the form of encoder, such as long-short term memory (LSTM) [4], transformer [5], and bidirectional encoder representations from Transformers (BERT) [6], have accelerated the blossom of text classification.

Currently, DNN-based text classification systems have obtained remarkable achievements, which mainly rely on large amounts of labeled data. For DNN-based supervised machine learning, there is a fundamental assumption that the training and testing data should be independently and identically distributed samples drawn from the same distribution. A subtle shift between training and testing data distributions can make the model produce poor classification accuracy. Unfortunately, the availability of labeled data may vary largely among different domains. For certain popular domains, such as book, movie, and electronic product reviews, manually-annotated datasets have been made public, while for domains such as medical equipment reviews, there may exist few annotated data [7]. Moreover, text classification is a highly-domain dependent task. the same word in different domains may express different sentiments [8]. For example, in the domain of electronics product review, the word "easy" frequently conveys a positive sentiment (e.g., the air conditioner is easy to use). While in the domain of movie reviews, "easy" is usually regarded as a negative indicator (e.g., the ending of this novel is easy to guess). A text classifier trained on a specific domain is inclined to perform poorly on a different domain. It is well-known that humans can learn knowledge of one task by transferring knowledge from certain related tasks [9]. Inspired by this observation, it is of great significance to develop models that can leverage the available labeled data from related domains to

improve the classification accuracy of the target domain.

To address the aforementioned issue, there exist three different streams. The first one is the domain-agnostic method, which combines all available labeled data from existing domains as the training set, ignoring domain differences. Unfortunately, text classification is highly domain-dependent. This method fails to produce a satisfactory result [10]. The second one is to fine-tune large pre-trained language models [11], such as BERT [6] and OpenAI GPT [12], on the target domain. The large pre-trained language models can effectively express context information of a word compared to general word embedding and yield remarkable performance in various NLP tasks [6]. The third one is multi-domain text classification (MDTC) [13], which applies transfer learning techniques to simultaneously capture knowledge that can be shared among multiple domains. Transfer learning aims to transfer knowledge from related domains and tasks to the target domain. It has been used successfully across some applications of machine learning, from image classification [14] and semantic segmentation [15] to sentiment analysis [16] and pre-trained word embeddings [17].

In this thesis, we aim to address the problem of MDTC using transfer learning techniques, we argue that framing the training of MDTC models as transfer learning rather than standard supervised learning can help excavate new perspectives that will enable the learned feature representations sharing across multiple domains and different categories. To this end, we employ adversarial learning to perform domain alignment. We demonstrate that our proposed models outperform state-of-the-art approaches and provide a generalization bound to close the gap between MDTC algorithms and theories.

1.2 Research Problems

This thesis studies the MDTC task. The main hypothesis of this thesis is the following: DNN-based MDTC models that leverage the relationship among the multiple domains outperform models ignoring domain differences. To explicitly model the relationship across different domains and exploit both labeled and unlabeled data, adversarial training, shared-private paradigm, and entropy minimization are encouraged to use in training MDTC models. Adversarial training is able to reduce divergence and perform alignment among multiple domains. It is often deployed between a feature extractor and a domain discriminator: the domain discriminator tries to distinguish features among multiple domains, while the feature extractor contrives to deceive the domain discriminator. When these two modules reach equilibrium, the generated features are supposed to be domain-invariant, which are both transferable and discriminative. In addition, shared-private paradigm is incorporated to excavate the domain-specific features to enhance the discriminability of the domain-invariant features [18]. Shared-private paradigm consists of two types of feature extractors: the shared feature extractor constructs the shared latent space to learn domain-invariant features, and a set of domain-specific feature extractors, one per domain, each of which captures domain-specific knowledge from its corresponding domain [19]. Entropy minimization is often employed in semi-supervised learning (SSL) to regularize the model output and minimize the uncertainty of predictions on unlabeled data [7, 20]. Nowadays, MDTC methods that adopt adversarial training, shared-private paradigm, and entropy minimization yield state-of-the-art performance. However, these methods still face some challenges. Here, we summarize these challenges in 6 aspects:

- How to explicitly model the relationship among the multiple domains.
- How to further reduce the domain divergence among domains in the latent space.

- How to use the limited amounts of data from each domain to guarantee the domain-invariance of the shared latent space.
- When performing adversarial training on marginal distributions, as the class-aware information is ignored, features that belong to different labels can be matched, resulting in weak discriminability of the learned features.
- Entropy minimization may encourage the model to make over-confident predictions on unlabeled data, deteriorating the discriminability of the learned feature representations
- There is no theoretical guarantees to support MDTC algorithms.

1.3 Contributions

Throughout this thesis, we will focus on addressing the aforementioned 6 challenges for MDTC methods. Here we enlist the specific contributions:

- We introduce a novel dual adversarial co-learning (DACL) approach which combines discriminator-based adversarial learning and classifier-based adversarial learning to refine the domain-invariant feature representations;
- We propose a dual mixup regularized learning (DMRL) method which introduces two forms of mixup-based regularizers to help reduce the distribution discrepancy among multiple domains and enforce consistent predictions in-between training samples.
- We propose a conditional adversarial network (CAN) that incorporates conditional domain discriminator and entropy conditioning to perform alignment on the joint distributions of shared features and label predictions to improve the system performance.

- We propose a co-regularized adversarial learning (CRAL) method which constructs two diverse shared latent spaces, performs adversarial alignment in each of them, penalizes the disagreements of their predictions on unlabeled data, and incorporates virtual adversarial training with entropy minimization to improve the system performance;

- We propose a margin discrepancy-based adversarial training (MDAT) method which employs the margin discrepancy to measure the domain divergence and guide the adversarial alignment among multiple distributions. Moreover, we close the gap between MDTC algorithms and theories.

1.4 Outline

The remainder of the thesis is organized as follows:

- Chapter 2 provides the essential background of adversarial learning and multi-domain text classification that are necessary for the understanding of our proposed approaches.

- Chapter 3 introduces the two most commonly used MDTC datasets and provides their details.

- Chapter 4 proposes a novel dual adversarial co-learning (DACL) approach for MDTC. This approach combines two adversarial learning mechanisms: the discriminator-based adversarial learning and the classifier-based adversarial learning to conduct alignment among different domains. The minimax optimization is deployed in an alternative fashion with three steps: the model learning step, the adversarial opponent step, and the model refinement step.

- Chapter 5 presents a mixup regularized adversarial network (MRAN) for MDTC. The proposed approach introduces two mixup-based interpolation regularizations: the category mixup regularization and the domain mixup regularization. More specifically, the category mixup regularization can enforce consistent predictions on the linear interpolations for both labeled and unlabeled samples, imposing stronger discriminability to the learned features. While the domain mixup regularization expands the searching range in the latent space, excavating more intrinsic feature patterns such that a more continuous latent space can be achieved.

- Chapter 6 proposes a conditional adversarial network (CAN), which is inspired by conditional generative adversarial networks, to explore the relationship between the shared feature representations and the label predictions to impose more discriminability on the learned features. The CAN approach introduces a conditional domain discriminator to model the domain variance in both shared features and class-aware information simultaneously and adopts entropy conditioning to guarantee the transferability of the learned features.

- Chapter 7 proposes a co-regularized adversarial learning (CRAL) mechanism for MDTC. This approach constructs two diverse shared latent spaces, performs domain alignment in each of them, and punishes the disagreements of these two alignments with respect to the predictions on unlabeled data. Moreover, virtual adversarial training (VAT) with entropy minimization is incorporated to impose consistency regularization to the CRAL method.

- Chapter 8 first provides a thorough theoretical analysis for incorporating the margin discrepancy to measure the domain divergence in MDTC and induce a new generalization bound based on the Rademacher complexity for MDTC. Then we propose a novel margin discrepancy-based adversarial training

(MDAT) method based on the theoretical analysis.

- Chapter 9 concludes this thesis and gives direction to future work.

1.5 Summary

Some of the work in this thesis appeared previously in the following publications:

1. Wu, Y., & Guo, Y. (2020, April). Dual adversarial co-learning for multi-domain text classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 6438-6445).

2. Wu, Y., Inkpen, D., & El-Roby, A. (2020, August). Dual mixup regularized learning for adversarial domain adaptation. In European Conference on Computer Vision (pp. 540-555).

3. Wu, Y., Inkpen, D., & El-Roby, A. (2021). Mixup Regularized Adversarial Networks for Multi-Domain Text Classification. In proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

4. Wu, Y., Inkpen, D., & El-Roby, A. (2021). Conditional Adversarial Networks for Multi-Domain Text Classification. In the Second Workshop on Domain Adaptation for NLP at the 16th Conference of the European Chapter of the Association for Computational Linguistics.

5. Wu, Y., Inkpen, D., & El-Roby, A. (2021). Towards Category and Domain Alignment: Category-Invariant Feature Enhancement for Adversarial Domain Adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 132-141).

6. Wu, Y., Inkpen, D., & El-Roby, A. (2022). Maximum Batch Frobenius Norm for Multi-Domain Text Classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

7. Wu, Y., Inkpen, D., & El-Roby, A. (2022). Co-Regularized Adversarial Learning for Multi-Domain Text Classification. In International Conference on Artificial Intelligence and Statistics. PMLR.

Chapter 2

Related Work

In this chapter, we first elaborate on the field of adversarial learning. Then we show the literature reviews on two fields that are mostly related to MDTC: (1) domain adaptation (Section 2.2); (2) multi-task learning. Finally, we will describe the multi-domain text classification problem and approaches

2.1 Adversarial Learning

Adversarial learning is first proposed in generative adversarial networks (GANs) [21], it trains a generator against a discriminator adversarially to generate fake images that are similar to real images: the discriminator tries to distinguish fake images from the real ones, while the generator strives to fool the discriminator. When these two components can not be improved, the generated images will be similar to the real ones. Adversarial learning is able to align the distributions of the fake images to that of the true images, it is often used to generate high-resolution images [22], produce adversarial examples for adversarial attack [23], and align different domains [24]. In this thesis, we focus on applying adversarial learning to perform domain alignment to generate domain-invariant features. In domain alignment, adversarial learning often uses Jensen-Shannon divergence to measure the domain divergence in

the latent space, it seeks to perform a minimax optimization to reduce discrepancy across different domains. The adversarial loss is commonly used in domain adaptation and MDTC, it can be added to various tasks to yield domain-invariant feature [16]. The domain-invariant feature is supposed to be both transferable and discriminative, it can generalize across different domains to conduct classification.

2.2 Domain Adaptation

Supervised machine learning is based on an assumption that the training and testing data should be independently and identically drawn from the same distribution. This assumption breaks down when the distributions of the training and testing data are different. Unfortunately, in many real-world applications, the availability of labeled data varies among different domains. It is impossible that every target domain contains sufficient amounts of labeled data to train an efficient classifier. This gives rise to domain adaptation (DA). DA is proposed to train models that learn knowledge from the source domain, which has abundant labeled data and generalize the learned knowledge to the target domain, which has scarce labeled data. DA has drawn tremendous attention in past decades [25]. There exists a more challenging task called unsupervised domain adaptation (UDA) where the target domain has no labeled data. In DA, the source and target domains share the same label space, but different data distributions. We now present the definition of DA following the notation of [25] with binary classification as a running example. A domain $\mathcal{D}$ consists of an input space $\mathcal{X}$, a label space $\mathcal{Y}$ and a marginal probability distribution $P(X)$ over the feature space, where $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \mathcal{X}$. For binary classification, $\mathcal{X}$ is the space containing all inputs, $\mathbf{x}_i$ is the i-th term vector, X is the random variable associated with the sample for training, and $\mathcal{Y}$ is the set of all labels, which is in $\{0, 1\}$. Given a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$, the objective of DA is to

learn the conditional probability distribution of the target domain $P_T(Y_T|X_T)$ with the knowledge learned from $\mathcal{D}_S$ where $\mathcal{D}_S \neq \mathcal{D}_T$.

The DNN-based DA methods project input data of both source and target domains into a low-dimensional latent space to obtain the source and target distributions, reduce the distance between source and target distributions based on a metric of domain similarity, and then capture domain-invariant features. The domain similarity is always quantified by the measurement of the distance between domains [25]. Here we briefly summarize the domain divergence measures from two perspectives: (1) Information theoretical measures; (2) Higher-order moment measures. Divergence measurement plays a significant role in DA models, it significantly influences the performance of the DA models [26]. The proliferation of divergence measures challenges researchers in choosing an appropriate measure for a given application. Here, we briefly review the most commonly used domain divergence measurements.

2.2.1 Information Theoretical Measures

Kullback-Leiberl (KL) divergence

The KL divergence is defined as follows:

$$D_{KL}(P_S \| P_T) = \mathbb{E}_{\mathbf{x} \sim P_S}[\log P_S(\mathbf{x}) - \log P_T(\mathbf{x})] \tag{2.1}$$

Where P_S is the source feature distribution and P_T is the target feature distribution. The KL divergence is a special case of an f-divergence [27], which is a general family of divergence measurements across different probability distributions.

Jensen-Shannon (JS) divergence

The JS divergence is a symmetric version of the KL divergence:

$$JSD(P_S \| P_T) = \frac{1}{2} D_{KL}(P_S \| M) + \frac{1}{2} D_{KL}(P + T \| M) \tag{2.2}$$

where $M = \frac{1}{2}(P_S + P_T)$

Rényi divergence

The Rényi divergence [28] is defined as:

$$D_R = \frac{1}{\alpha - 1} \mathbb{E}_{\mathbf{x} \sim P_S} [\log P_S(\mathbf{x})^{\alpha} - \log P_T(\mathbf{x})^{\alpha - 1}] \tag{2.3}$$

where $0 < \alpha < \infty$ and $\alpha \neq 1$. In the extreme case when $\alpha \mapsto 1$, the Rényi divergence is equivalent to the KL divergence.

Wasserstein distance

The Wasserstein distance [29] measures the minimum loss of transforming the source distribution into the target distribution. It is also known as the Earth Mover's distance. Its definition is:

$$W(P_S, P_T) = \max \mathbb{E}_{\mathbf{x} \sim P_S} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim P_T} [f(\mathbf{x})] \tag{2.4}$$

Where $f(\cdot)$ is the mapping function parameterized by DNN, and $f(\cdot)$ should be a 1-Lipschitz continuous function, i.e. a function with absolute value bounded by 1.

2.2.2 Higher-Order Moment Measures

Maximum Mean Discrepancy (MMD)

MMD [30, 31] aims to measure the distance between the sample means of the source and target distributions P_S and P_T in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. The empirical estimate of the MMD is defined as:

$$MMD(P_S, P_T) = \|\mathbb{E}_{\mathbf{x}\sim P_S}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_T}[f(\mathbf{x})]\|_{\mathcal{H}} \tag{2.5}$$

Where $f(\cdot) : \mathcal{X} \mapsto \mathcal{H}$, i.e. $f(\cdot)$ maps an instance $\mathbf{x}$ to the RKHS. In practice, the mapping is obtained using an appropriately chosen kernel such as an RBF kernel [18] or a DNN [32]. In this case, MMD can be seen as a measurement of the distance between the means of embeddings of X_S and X_T.

Proxy $\mathcal{A}$ distance

The $\mathcal{A}$ distance [33] is defined as follows:

$$d_{\mathcal{A}}(P_S, P_T) = 2 \sup_{A\in\mathcal{A}} |P_S(A) - P_T(A)| \tag{2.6}$$

Where $\mathcal{X}$ represents a feature space and $\mathcal{A}$ is a collection of subsets of $\mathcal{X}$. Intuitively, the $\mathcal{A}$ distance is the largest change in probability of a subset from the source distribution to the target distribution. Moreover, calculating the $\mathcal{A}$ distance for finite samples is equivalent to minimizing the error rate of a domain discriminator that is trained to distinguish source features from target features [34]. Practically, the proxy $\mathcal{A}$ distance is approximated as:

$$pd_{\mathcal{A}}(P_S, P_T) = \frac{100}{|X_S| + |X_T|} \sum_{\mathbf{x},y\in X_S+X_T} 1 - \mathcal{L}_\delta(\mathbf{x}, y) \tag{2.7}$$

Where $\mathcal{L}_\delta$ is the Huber loss and $y = 1$ if $\mathbf{x} \in X_S$ and $y = 0$ if $\mathbf{x} \in X_T$. If the loss is large, the proxy $\mathcal{A}$ distance is close to 0 or negative, which indicates that the domains are indistinguishable using a linear classifier. If the domains are maximally different, the loss is 0, and the proxy $\mathcal{A}$ distance is 100; a classifier is thus easily able to tell them apart.

Correlation Alignment (CORAL)

CORAL aims to minimize domain divergence by aligning the second-order statistics of the source and target distributions [35]. Its definition is:

$$CORAL(C_S, C_T) = \frac{1}{4d^2}||C_S - C_T||_F^2 \tag{2.8}$$

Where d is the feature representation dimension, C_S and C_T denote the covariance matrices of the source and target data, $||\cdot||_F$ denotes the Frobenius norm. [36] extends CORAL to DNNs with a linear transformation.

Central Moment Disrepancy (CMD)

CMD considers not only the first and second moments, but also higher-order moments [37]. Assume each variable in P_S and P_T is bounded by $[a, b]$, the approximation of the CMD is defined as:

$$CMD_K(P_S, P_T) = \frac{1}{(b-a)}||E(X_S) - E(X_T)||_2 + \sum_{K=2}^{\infty}\frac{1}{|b-a|^K}||c_k(X_S) - c_k(X_T)||_2 \tag{2.9}$$

Where $E(\cdot)$ is the expectation and $c_k(\cdot)$ is the k-th order central moment, defined as:

$$c_k(X) = E(\prod_{i=1}^{N}(X_i - E(X_i))^{r_i}) \tag{2.10}$$

Where N is the number of samples in X, $\sum_{i=1}^{N} r_i = N$, and $r_1, ..., r_N \geq 0$.

For DA methods, the most common strategy is to minimize the classification loss on the source domain so that the learned knowledge can be distinguishable across categories, and minimize the divergence between the source and target domains such

that the learned knowledge can be transferable across domains. In the following, we give the literature reviews on two main types of DA methods:

- statistical-based approaches
- adversarial learning-based approaches

These two DA streams both focus on capturing domain-invariant feature representations via minimizing the divergence between the source and target domains. DA has four different settings: (1) single source & single target; (2) multiple sources & single target; (3) single source & multiple targets; (4) multiple sources & multiple targets. Here we only focus on the most common single source & single target setting.

2.2.3 Statistical-based Approaches

Statistical-based DA approaches directly use domain similarity as the measure to reduce the domain divergence in the latent space. [38] first introduces the MMD metric in the DA setting. The deep domain confusion (DDC) method uses two convolutional neural networks with shared weights for the source and target domains and leverages MMD in the last fully-connected layer to learn domain-invariant features [32]. The deep adaptation network (DAN) minimizes multi-kernel MMD (MK-MMD) in several task-specific layers to achieve advanced success [39]. The joint adaptation network (JAN) employs a joint MMD criterion to conduct joint distribution alignment on multiple domain-specific layers to yield domain-invariant features [40]. [35] first proposes CORAL to match the mean and covariance of the two distributions. Then [36] extends CORAL to DNN to propose Deep CORAL. [41] uses a Euclidean distance to propose mapped CORAL. [42] combines log-Euclidean distance and CORAL to propose LogCORAL. [43] generalizes CORAL to possibly infinite-dimensional covariance matrices in RHKS. [37] first proposes CMD to match the higher-order moments

of the source and target distributions. [44] proposes contrastive domain discrepancy (CCD) based on MMD to reduce divergence on conditional distributions of the source and target domains, CCD seeks to minimize intra-class discrepancy while maximizing inter-class discrepancy.

2.2.4 Adversarial Learning-based Approaches

In recent years, with the inspiration of GAN [21], the domain adversarial neural network is proposed [45]. It begins the investigations of adversarial training-based DA methods and follows the principle that using representations to minimize the discrepancy between the source and target feature distributions while maximizing the performance on the source domain data [24]. Minimizing the error rate on the source data enables the feature representations to be discriminative while minimizing the discrepancy between the source and target feature distributions enable the feature representations to be transferable. The transferability is often obtained through a minimax game between a feature extractor and a domain discriminator: the domain discriminator tries to distinguish source features from the target features while the feature extractor strives to deceive the domain discriminator. When the domain discriminator can not identify the differences between the source and target features, the feature extractor can capture the domain-invariant features. The adversarial loss is defined as:

$$\mathcal{L}_{ADV}(P_S, P_T) = \mathbb{E}_{x\sim P_S}[\log D(G(x))] + \mathbb{E}_{z\sim P_T}[\log(1 - D(G(z)))] \tag{2.11}$$

Where $D(\cdot)$ is the mapping function of the domain discriminator, $G(\cdot)$ is the mapping function of the feature extractor. The training and testing processes of DANN are illustrated in Fig. 1.

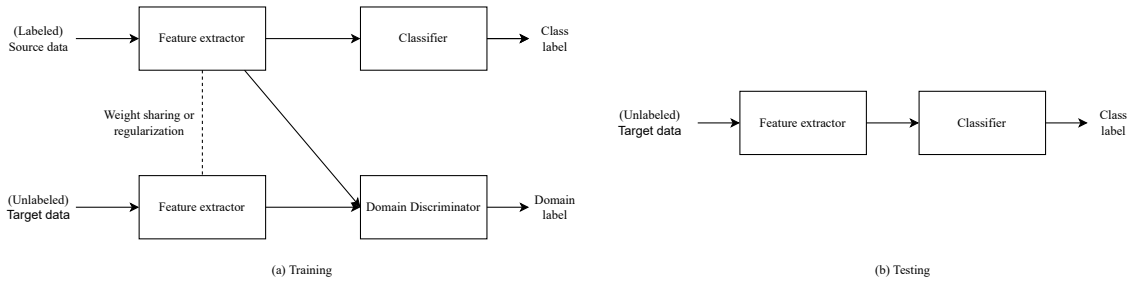


Figure 1: The training and testing processes of the DANN.

The adversarial discriminative domain adaptation (ADDA) method uses an untied weight sharing strategy to align the feature distributions of the source and target domains [46]. The coupled GAN (CoGAN) consists of two GANs to generate source and target data respectively, ties the weights of the first few layers of the generators and the last few layers of the discriminators to achieve a domain-invariant latent space without corresponding supervision [47]. The domain separation network introduces the shared-private paradigm, which incorporates the domain-specific features to enhance the discriminability of the domain-invariant features [18]. [48] performs binary classification via an additional domain discriminator layer and introduces a domain confusion loss to encourage its prediction to be close as possible to a uniform distribution over binary labels. The maximum classification discrepancy (MCD) proposes the task-specific classifier-based adversarial training between a feature extractor and two classifiers to learn domain-invariant features [49]. The cycle-consistent adversarial domain adaptation (CyCADA) employs cycle-consistent adversarial training to perform domain alignment at both pixel and feature levels [50]. The multi-adversarial domain adaptation (MADA) exploits multiple discriminators to enable multiplicative interactions between features and labels to improve adversarial training [51]. The generate to adapt (GTA) performs joint distribution alignment by introducing an additional image generation module [52]. The conditional domain adversarial network (CDAN)

updates the domain discriminator by using multilinear maps of features and label predictions as inputs so as to enable conditional distribution alignment [20]. The consensus adversarial domain adaptation (CADA) method seeks agreement between the source and target encoders to reach a domain-invariance latent space [53]. The wasserstein distance guided representation learning (WDGRL) method uses wasserstein distance as the measurement to guide the domain discriminator in the adversarial learning to learn domain-invariant features [54]. The transferable prototypical network (TPN) aims to align three distribution metrics of each class between the source and target domains [55]. The batch spectral penalization (BSP) method improves the feature discriminability by imposing restrictions on the eigenvalues of the domain-invariant features [56]. The enhanced transport distance (ETD) method exploits attention-aware transport distance and entropy-based regularization to advance the discriminability of the domain-invariant features [57].

2.3 Multi-task learning

Machine learning algorithms conventionally aim to solve a single specific task, it ignores information from other related tasks that might help improve the performance of the target task. Multi-task learning (MTL) is proposed to use one model to tackle many tasks simultaneously. In recent decades, it has drawn much attention from various communities as it leads to successes in various applications of machine learning, from NLP [58] to computer vision [59]. MTL learns shareable feature representations that can generalize across different tasks [60]. The motivation of MTL is that when humans learn to tackle unseen tasks, we often use the knowledge we acquired from related tasks. For example, a baby first learns to recognize faces and can then apply this knowledge to recognize other objects [61].

In MTL, assume we have m different learning tasks $\{T_i\}_{i=1}^{m}$, all tasks are supposed to be related. Each task is accompanied by a dataset D_i whose distribution is P_i. Each D_i contains n_i training samples, i.e., $D_i = \{\mathbf{x}_j^i \in \mathbb{R}^{d_i}, y_j^i\}_{j=1}^{n_i}$, where d_i is the dimension of the data vector of D_i. We use $X^i = (\mathbf{x}_1^i, ..., \mathbf{x}_{n_i}^i)$ to denote the training data matrix for T_i. MTL assumes that all tasks share the same input space where $d_i = d_j$ for any $i \neq j$. In earlier studies of MTL, task-relatedness is supposed to be prior information. [62] assumes all tasks are similar to each other such that the model parameters of each task are enforced to approach the average model parameters of all tasks. [63] regularizes the model parameters in the principle that the more similar two tasks are, the closer the corresponding model parameters are expected to be. However, The relationships across different tasks are often unavailable in practice, using task similarity, task correlation, and task covariance to quantify the task relatedness becomes the mainstream of MTL [64]. [65] learns the task similarity as well as the number of clusters, sharing the same model among tasks in the same cluster. [66] proposes a hierarchical bayesian model, learning a latent task hierarchy. [67] uses a gaussian process-based regularizer to enable the model to be more computationally feasible. [68] assumes that tasks form several disjoint groups and learn a model per group with a group assignment matrix. [69] divides all tasks into multiple disjoint groups, assumes there exist a small number of basic tasks, and enables two tasks from different groups to overlap. [70] maps each task to a single hypothesis and assumes these hypotheses can overlap to some extent. The deep relationship network (DRN) employs shared and task-specific layers within the model and uses matrix priors to advance the performance [39]. The fully-adaptive feature sharing method dynamically adds branches to the model when encountering new tasks, it uses a greedy strategy to promote the grouping of similar tasks [71]. The cross-stitch network uses cross-stitch units to determine in what way the task-specific networks leverage the knowledge from other tasks by learning a linear combination

of the output of the previous layer [72]. The multi-gate mixture-of-experts method contains multiple shared encoders and each task-specific output head receives a linear combination of the outputs of each encoder [73]. The pattern affinitive propagation method employs an affinity learning layer to capture pair-wise relationships of tasks and combines features from different tasks based on the relationships [74].

2.4 Multi-domain text classification

Multi-domain text classification (MDTC) can be regarded as a combination of domain adaptation and multi-task learning. It simultaneously performs domain alignment among multiple domains to learn shareable feature representations. Currently, there are two major streams for MDTC approaches: (1) fully-shared paradigm [75]; (2) shared-private paradigm [10, 19]. In the fully-shared paradigm, different domains share a single feature extractor within the model. This type of model only considers shareable features across domains, ignoring the domain-specific knowledge. Figure 2(a) illustrates the fully-shared model. In the shared-private paradigm [18], as illustrated in Figure 2(b), two different feature spaces are introduced: the shared latent space seeks to capture the domain-invariant features across different domains, while the domain-specific latent space, each for per domain, contrives to capture domain-dependent features. The shared-private paradigm demonstrates that domain-specific information can boost the discriminability of the domain-invariant features.

Many recent MDTC methods adopt the shared-private paradigm. Meanwhile, in order to explicitly model the separations of the shared latent space and the multiple domain-specific latent spaces, preventing the domain-specific feature from creeping into the shared latent space, adversarial training is also incorporated in MDTC. By deploying adversarial training between a domain discriminator and a shared feature extractor, the total divergence among multiple feature distributions can be minimized.

In adversarial training, the domain discriminator is trained to distinguish features among multiple domains, while the shared feature extractor learns to fool the domain discriminator. The adversarial multi-task learning for text classification (ASP-MTL) utilizes long short-term memory (LSTM) without attention as feature extractors and orthogonality constraints to help separate domain-invariant and domain-specific features to encode different aspects of the inputs [19]. The dynamic attentive sentence encoding (DA-MTL) method utilizes the attention mechanism and introduces a task-dependent query vector to learn domain-specific features to enhance the discriminability of the shared features [11]. The multinomial adversarial network (MAN) derives the convergence conditions of MDTC in terms of two loss functions: the least square loss (MAN-L2) and the negative log-likelihood loss (MAN-NLL) [10]. The multi-task learning with bidirectional language models for text classification (MT-BL) introduces language modeling as an auxiliary task to encourage the domain-specific feature extractors to capture more syntactic and semantic information, and a uniform label distribution-based loss constraint to the shared feature extractor to enhance the ability to learn domain-invariant features [76]. The global and local shared representation based generic dual-channel multi-task learning (GLR-MTL) method deploys adversarial training and mixture-of-experts on two separate channels to capture global-shared, local-shared, and private features simultaneously [77]

In MDTC, the main task is sentiment analysis, which is the task of classifying the polarity of a given text. Usually, this polarity is binary (positive or negative). Most MDTC datasets contain domains with a limited amount of labeled data and a large amount of unlabeled data, the data are all emotive text data, such as movie and product reviews. In these reviews, star ratings are generally used as a proxy for sentiment. Sentiment analysis has become one of the most popular tasks in NLP and many methods have been developed for sentiment analysis [78]. Variants of the sentiment analysis task employ different notions of sentiment and require determining

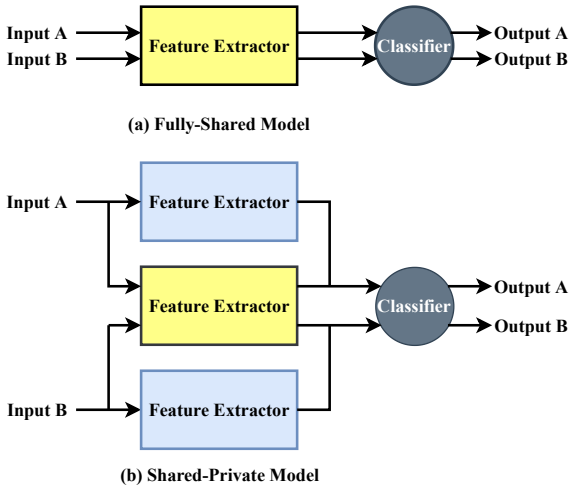


Figure 2: Two paradigms for tackling MDTC tasks.

the sentiment with respect to a particular target.

2.4.1 Task formulation

In multi-domain text classification, we assume there exist M domains, each of which contains a limited number of labeled instance $\mathbb{L}_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^{l_i}$, and a set of unlabeled instances $\mathbb{U}_i = \{\mathbf{x}_j\}_{j=1}^{u_i}$, where l_i and u_i are numbers of labeled and unlabeled samples, respectively. We are required to learn feature representations from all domains and use the representations to conduct classification across all domains. The system performance is measured as the average classification accuracy across the M domains. The challenge of MDTC lies in how to improve the system performance by leveraging all available resources across different domains.

2.4.2 Evaluation metrics

MDTC systems are typically evaluated with regard to their performance on the test set of the specific task. For binary classification, accuracy is the common evaluation measure, which is defined as follows:

$$Acc = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.12}$$

Where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives respectively. Intuitively, the number of true predictions is divided by the number of all predictions.

2.5 Summary

In this chapter, we have introduced background knowledge in domain adaptation, multi-task learning, and multi-domain text classification. In the domain adaptation section, we also elaborate on several most commonly used divergence measurements. In the next chapter, we will describe the most popular datasets used to evaluate the performance of multi-domain text classification models and present their details.

Chapter 3

Dataset

Having a comprehensive and reliable dataset is a critical factor for training a well-performed model for machine learning. This chapter provides descriptions of the MDTC datasets used in this thesis.

3.1 The Amazon Review Dataset

The Amazon review dataset [79] consists of four domains: books, DVDs, electronics, and kitchen appliances. For each domain, there exist 2,000 instances: 1,000 positive ones and 1,000 negative ones. All data was pre-processed into a bag of features (unigrams and bigrams), losing all word order and semantic information. The pre-processed data can not be directly processed by powerful encoders, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer. We hence use multi-layer perceptrons (MLPs) as feature extractors for experiments on the Amazon review dataset. In the experiments, the 5,000 most frequent features are used, representing each review as a 5,000-dimensional vector. Since the Amazon review dataset is rather small, when conducting experiments on it, we follow the standard MDTC experimental setting [10, 16]: a 5-fold cross-validation is implemented: All data is divided into 5 folds per domain, three of the five folds are used as the

Table 1: Statistics of the Amazon review dataset

Domain	Labeled	Positive	Negative	Unlabeled	Class.
Books	2000	1000	1000	4465	2
Electronics	2000	1000	1000	3586	2
DVD	2000	1000	1000	568	2
Kitchen	2000	1000	1000	5945	2

training set, one is the validation set, and the remaining one is treated as the test set. The 5-fold average test accuracy is reported. The detailed statistics of the Amazon review dataset are given in Table 1

3.2 The FDU-MTL Dataset

Although the Amazon review dataset is widely adopted as benchmark in MDTC, it has two limitations: (1) It only has four domains; (2) The data are already tokenized and converted to bags of features, making it impossible to use powerful networks, such as CNNs or RNNs, as feature extractors to capture feature representations. To comprehensively evaluate the effectiveness of the MDTC approaches, we use the FDU-MTL dataset in the experiments. The FDU-MTL dataset [19] is a more challenging dataset, containing 16 domains: books, electronics, DVDs, kitchen, apparel, camera, health, music, toys, video, baby, magazine, software, sport, IMDB, and Movie Reviews (MR). The first 14 domains are Amazon reviews on different products and the other two domains are movie reviews. All the data are collected as raw text data only being tokenized by the Stanford tokenizer [80]. Thus, this dataset allows one to use powerful feature extractors. The data in each domain is randomly divided into three parts: training set (70%), validation set (10%), and testing set (20%). There exist 200 samples in the validation set and 400 samples in the testing set for each domain.

While the numbers of labeled and unlabeled instances vary across domains they are roughly 1400 and 2000, respectively. The detailed statistics of the FDU-MTL dataset are listed in Table 2.

3.3 Summary

In this chapter, we present two benchmarks for multi-domain text classification and their details. In the next chapter, we will discuss the dual adversarial co-learning method for multi-domain text classification.

Domain	Train	Dev.	Test	Unlabeled	Avg. L	Vocab.
Books	1400	200	400	2000	159	62K
Electronics	1398	200	400	2000	101	30K
DVD	1400	200	400	2000	173	69K
Kitchen	1400	200	400	2000	89	28K
Apparel	1400	200	400	2000	57	21K
Camera	1397	200	400	2000	130	26K
Health	1400	200	400	2000	81	26K
Music	1400	200	400	2000	136	60K
Toys	1400	200	400	2000	90	28K
Video	1400	200	400	2000	156	57K
Baby	1300	200	400	2000	104	26K
Magazine	1370	200	400	2000	117	30K
Software	1315	200	400	475	129	26K
Sports	1400	200	400	2000	94	30K
IMDB	1400	200	400	2000	269	44K
MR	1400	200	400	2000	21	12K

Table 2: Statistics of the FDU-MTL dataset

Chapter 4

Dual Adversarial Co-Learning for Multi-Domain Text Classification

4.1 Introduction

With the advent of deep learning, natural language processing (NLP) has drawn much attention in past years and obtained remarkable accomplishments [81]. Text classification is a fundamental task in NLP, it is well-known to be a highly domain-dependent task [82]. A subtle bias between the training and test data distributions can cause catastrophic performance deterioration. Unfortunately, the availability of labeled data may vary across different domains. Collecting a sufficient amount of labeled data for target domains is often expensive and time-consuming, Therefore, it is of great significance to investigate how to improve the classification accuracy of the target domain by leveraging available resources from related domains.

A simple solution to address the above problem is to combine all labeled data from all domains together as the training set to train a classifier, ignoring domain differences. As presented in 2.4, text classification is a highly domain-dependent task, such methods can not yield satisfactory results. Multi-domain text classification (MDTC) is proposed to consider the domain relatedness [79] in training classifiers.

MDTC seeks to simultaneously utilize all available data across multiple domains to improve the classification accuracy of each domain. Recently, DNN-based models have been widely used for multi-domain learning in both computer vision [83, 84] and NLP [10, 85]. Existing MDTC approaches train to obtain shareable feature representations by sharing model parameters across different domains [75]. However, focusing only on a shared latent space across different domains may omit the domain-specific characteristics of each domain. One way to address this issue is to employ the shared-private paradigm to consider the domain-specific features per domain in training classifiers [18]. The shared-private paradigm suffers from problems such as domain-dependent features that may infiltrate into shared latent space, resulting in performance deterioration [19]. Moreover, with the reality of lacking sufficient labeled data in MDTC to train an efficient classifier, the learned features may overfit the labeled training data and fail to generalize well to other domains.

To address the aforementioned problem, in this chapter, we propose a novel dual adversarial co-learning (DACL) approach for MDTC. The approach captures both domain-invariant and domain-specific features through the shared-private paradigm and trains two classifiers on the learned features. the DACL approach combines the standard discriminator-based adversarial training [45] and the classifier-based adversarial training [49] and co-trains the feature extractors against the discriminator and two classifiers in an alternative fashion [21]. In particular, the feature extractors and the two classifiers are adversarially trained based on the prediction discrepancy of the unlabeled data, while the shared feature extractor and the domain discriminator are trained against each other to obtain domain-invariant features as well as separating the shared latent space from the domain-specific latent spaces. Different from previous MDTC approaches, our proposed DACL method not only aligns feature distributions across different domains in the shared latent space, but also simultaneously aligns labeled and unlabeled data per domain, avoiding overfitting to the limited labeled

data. The co-learning mechanism is implemented through three adversarial steps, seeking to yield feature representations that both generalize well across domains and into unlabeled data. We evaluate the efficacy of the proposed DACL method on the two MDTC benchmarks, we also extend the DACL approach for multi-source unsupervised domain adaptation (MS-UDA), demonstrating that our DACL method has the ability to generalize to unseen domains. The experimental results reveal that our DACL method achieves state-of-the-art performance both in MDTC and MS-UDA settings. Our contributions are listed below:

- We propose a dual adversarial co-learning (DACL) method for MDTC. The DACL method combines two forms of adversarial learning: the discriminator-based adversarial learning and the classifier-based adversarial learning, and uses the orthogonal regularizer to prevent the domain-specific features from creeping into the shared latent space.
- We evaluate the effectiveness of the proposed DACL method on two MDTC benchmarks, the experimental results demonstrate that our DACL method can outperform the state-of-the-art approaches.
- We conduct further analyses to see how each component in DACL can contribute to the whole model and how each hyperparameter can influence the system performance.

4.2 Dual Adversarial Co-Learning

As MDTC needs to tackle multiple domains and there only exists a limited amount of labeled data in each domain, to effectively capture domain-invariant features and

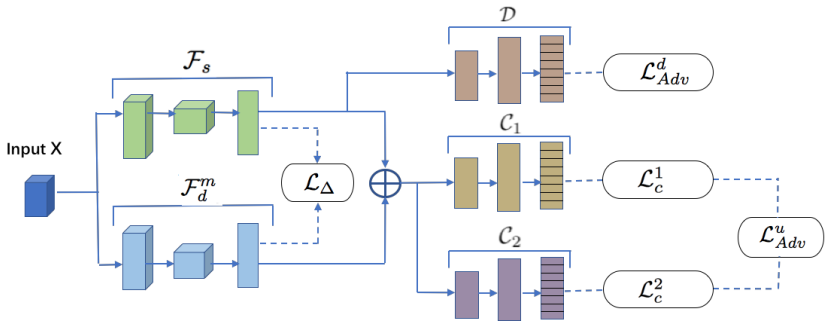


Figure 3: Architecture of the dual adversarial co-learning model.

avoid overfitting, our DACL approaches need to not only align different feature distributions in the shared latent space, but also align labeled and unlabeled data distributions. The architecture of the DACL approach is illustrated in Figure 3. The DACL model has five component networks: a shared feature extraction network $\mathcal{F}_s$, M domain-specific feature extraction networks $\{\mathcal{F}_d^m\}_{m=1}^M$, two classification networks $\mathcal{C}_1$ and $\mathcal{C}_2$, and a domain discriminator $\mathcal{D}$. The shared feature extractor $\mathcal{F}_s$ learns to capture the shared features that can generalize across all domains, while each domain-specific feature extractor $\mathcal{F}_d^m$ learns to capture domain-dependent features that only contribute specifically to their own domain. These feature extractors perform data representation learning and the forms of the feature extractors are flexible, they can adopt Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), or Multiple Layer Perceptron (MLP), depending on the particular task. Each input instance will go through the shared feature extractor and the corresponding domain-specific feature extractor to generate two feature vectors for the consequent classification and domain discrimination.

The proposed DACL method combines two forms of adversarial learning mechanisms to generate useful and generalizable features that not only align multiple feature distributions but also align labeled and unlabeled data per domain through co-learning of a pair of classifiers. We introduce each of the adversarial learning

mechanisms below.

4.2.1 Domain Discrimination based Adversarial Learning

In the proposed DACL method, the shared feature extractor $\mathcal{F}_s$ and the domain-specific feature extractors $\{\mathcal{F}_d^m\}_{m=1}^M$ are expected to complement each other and maximally capture useful feature representations for multiple domains.

The domain-invariant features are the domain generalizable features that enable knowledge sharing across domains. In MDTC, it is important to learn domain-invariant features with sufficient discriminability and prevent the domain-invariant and domain-specific features from interfering with each other. To achieve this goal, we deploy a multinomial domain discriminator $\mathcal{D}$ to distinguish features among multiple domains in the shared latent space. For M domains, we set $\mathcal{D}$ to be a M-class classifier that takes the domain-invariant feature vector as input and outputs a probability vector on each input vector; e.g., $\mathcal{D}_m(\mathcal{F}_s(\mathbf{x}))$ denotes the predicted probability of instance $\mathbf{x}$ coming from the m-th domain. Intuitively, the multinomial discriminator $\mathcal{D}$ should be optimized to maximally identify features from multiple domains, while the shared feature extractor $\mathcal{F}_s$ contrives to fool the discriminator. The intuition is that if a strong discriminator can not tell the original domains of the input instances, the yielded features are essentially domain-invariant. We encode the multinomial adversarial loss as follows:

$$\min_{\mathcal{F}_s} \max_{\mathcal{D}} \quad \mathcal{L}_{Adv}^d = \sum_{m=1}^{M} \mathbb{E}_{\mathbf{x}\sim\mathbb{L}_m\cup\mathbb{U}_m} \log[\mathcal{D}_m(\mathcal{F}_s(\mathbf{x}))] \tag{4.1}$$

This discriminator-based adversarial loss extends the standard binary adversarial loss into the multi-domain learning scenarios under the same adversarial training principle.

4.2.2 Separation regularizer

Inspired by some recent works that adopt shared-private paradigm [18,19] on network separation analysis, to prevent the domain-specific knowledge from infiltrating into the shared latent space, we incorporate a separation regularizer to guarantee that the shared and domain-specific feature extractors encode different aspects of the input data.

In particular, the separation regularizer is used to minimize the similarity between the learned domain-invariant features and the domain-specific features such that they will not interfere with each other:

$$\min_{\mathcal{F}_s,\{\mathcal{F}_d^m\}} \mathcal{L}_\Delta = \sum_{m=1}^{M} \Big\| \sum_{\mathbf{x}\sim\mathbb{L}_m} \mathcal{F}_s(\mathbf{x})\mathcal{F}_d^m(\mathbf{x})^\top \Big\|_F^2 \tag{4.2}$$

where $\mathcal{F}_s(\mathbf{x})$ and $\mathcal{F}_d^m(\mathbf{x})$ denote the column domain-invariant and domain specific feature vectors, respectively. $\|\cdot\|_F$ denotes the Frobenius norm. By minimizing such a separation regularizer, the difference between the shared latent space and each domain-specific space will be maximized.

4.2.3 Prediction Discrepancy based Adversarial Learning

After feature extraction, the two classifiers $\mathcal{C}_1$ and $\mathcal{C}_2$ are deployed to take the concatenation of shared and domain-specific features as input and output the class probability of the given instance. The networks $\mathcal{C}_1$ and $\mathcal{C}_2$ can be MLPs that use a softmax output layer to produce prediction probabilities of each class. To impose discriminability to the learned features, both classifiers are trained together with the feature extractors by minimizing the cross-entropy loss on the labeled data:

$$\min_{\mathcal{C}_1,\mathcal{C}_2,\mathcal{F}_s,\{\mathcal{F}_d^m\}} \quad \mathcal{L}_c^1 + \mathcal{L}_c^2,$$
$$\text{where} \quad \mathcal{L}_c^i = -\sum_{m=1}^{M} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{L}_m} \log[\mathcal{C}_{iy}(\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^m(\mathbf{x}))],$$

$i \in \{1, 2\}$ denotes the index of two classifiers, and $\mathcal{C}_{iy}(\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^m(\mathbf{x}))$ denotes the prediction probability of instance $\mathbf{x}$ belonging to the y-th class by the classifier $\mathcal{C}_i$, we use the negative log-likelihood (NLL) to encode the loss function. Then $\mathcal{L}_c^1$ and $\mathcal{L}_c^2$ denote the classification losses of the two classifiers on the labeled data of M domains respectively. As there exist limited amounts of labeled data in each domain for MDTC, to avoid overfitting of the features to the labeled data and guarantee they can generalize to unlabeled data, we employ the classifier-based adversarial learning which uses the prediction discrepancy of the two classifiers on unlabeled data to assist feature representation learning, especially for the domain-specific features:

$$\min_{\mathcal{F}_s,\{\mathcal{F}_d^m\}} \max_{\mathcal{C}_1,\mathcal{C}_2:d(\mathcal{C}_i,\mathcal{C}_i^*)\leq\epsilon} \quad \mathcal{L}_{Adv}^u \tag{4.3}$$
$$\text{where} \quad \mathcal{L}_{Adv}^u = \sum_{m=1}^{M} \mathbb{E}_{\mathbf{x}\sim\mathbb{U}_m} \left\| \mathcal{C}_1(\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^m(\mathbf{x})) - \mathcal{C}_2(\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^m(\mathbf{x})) \right\|_1 \tag{4.4}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm function and we use ℓ_1 norm to measure the prediction discrepancy between the outputs of the two classifiers on unlabeled data, $d(\cdot, \cdot)$ denotes a distance function, and $\mathcal{C}_i^*$ $(i = 1, 2)$ denotes the classifiers trained on labeled data. The constraint $d(\mathcal{C}_i, \mathcal{C}_i^*) \leq \epsilon$ regularizes the parameters of the classifiers to enforce them to be in the close neighborhood of the original classifiers $\mathcal{C}_1^*$ and $\mathcal{C}_2^*$. The classifier-based adversarial training mechanism induces features that are robust to small changes in the classifiers. By maximizing the prediction discrepancy of the two classifiers $(\mathcal{C}_1, \mathcal{C}_2)$ on unlabeled data in the close neighborhood of the original classifiers (within distance ϵ), the features of unlabeled data that are not of the scope

of or not aligned with the features of the labeled data in the shared latent space can be identified. These unlabeled data are most likely located near the classification boundary of the original classifiers $(\mathcal{C}_1^*, \mathcal{C}_2^*)$, some near neighbor classifiers $(\mathcal{C}_1, \mathcal{C}_2)$ are inclined to make false predictions on them. With such an adversarial opponent, the feature extractors can be adapted to yield feature representations in which unlabeled data can be well aligned with labeled data per domain by minimizing the prediction agreement of unlabeled data, we can hence impose better generalization ability to the classifiers.

4.2.4 Stepwise Adversarial Training

To integrate the two adversarial training mechanisms introduced above into the training process of our DACL approach, we propose a step-wise adversarial training procedure, which has three sequential steps: model learning step, adversarial opponent step, and model refinement step. These three steps work together to implement robust domain adaptation, we present the three steps below.

Model Learning Step The proposed DACL model mainly comprises the feature extractors, $\mathcal{F}_s$ and $\{\mathcal{F}_d^m\}$, and the two classifiers, $\mathcal{C}_1$ and $\mathcal{C}_2$. We first train them to impose discriminability to the learned features by minimizing the classification loss on labeled data in multiple domains:

$$\min_{\mathcal{F}_s, \{\mathcal{F}_d^m\}, \mathcal{C}_1, \mathcal{C}_2} \mathcal{L}_c^1 + \mathcal{L}_c^2 + \alpha \mathcal{L}_\Delta \tag{4.5}$$

where α is a trade-off hyperparameter. We use the separation regularizer here to avoid the negative effect brought by the interference between domain-invariant and domain-specific features. This step only performs standard supervised training, the model needs further refined through adversarial training to impose transferability to

the learned features.

Adversarial Opponent Step We then combine the two adversarial training shown in Eq.(4.1) and Eq.(4.4) to perform alignment. Instead of implementing an alternating minimax optimization, we simply conduct two discrete steps: Maximizing the opponents (discriminator and classifiers) in the first step and minimizing the feature extractors in the second step. To avoid using restrictions to limit the search range of the classifiers within a close neighborhood of the initial classifiers in the latent space, we reformulate the adversarial objective by taking the classification loss on labeled data into consideration:

$$\max_{\mathcal{C}_1,\mathcal{C}_2} \quad -\left(\mathcal{L}_c^1 + \mathcal{L}_c^2\right) + \mathcal{L}_{Adv}^u \tag{4.6}$$

$$\max_{\mathcal{D}} \quad \mathcal{L}_{Adv}^d \tag{4.7}$$

These adversarial maximizations identify the difficulties in aligning domains in the shared latent space and aligning labeled and unlabeled features in the joint latent spaces respectively, which provide directions for further refining parameters of the feature extractors in the next step.

Model Refinement Step In the last step, we aim to refine the feature extractors ($\mathcal{F}_s$ and $\{\mathcal{F}_d^m\}$) to deceive the discriminator and minimize the prediction discrepancy of the outputs of the two classifiers on unlabeled data:

$$\min_{\mathcal{F}_s,\{\mathcal{F}_d^m\}} \quad \mathcal{L}_{Adv}^u + \gamma \mathcal{L}_{Adv}^d \tag{4.8}$$

The refined features are expected to be more robust and generalizable across all domains.

We adopt a mini-batch stochastic gradient descent algorithm for this step-wise

adversarial training, which is illustrated in Algorithm 1. In each iteration of the algorithm, the three steps, L-step (model Learning step), A-step (Adversarial opponent step), and R-step (model Refinement step), should be implemented sequentially over the sampled batches, while each step performs a stochastic gradient-based update regarding the related objective.

Algorithm 1 Stochastic gradient descent training algorithm

1: **Input:** labeled data $\mathbb{L}_m$ and unlabeled data $\mathbb{U}_m$ in M domains; hyper-parameters α, γ.

2: **for** number of training iterations **do**

3: Sample labeled mini-batches from the multiple domains $B^\ell = \{B_1^\ell, \cdots, B_M^\ell\}$.

4: Sample unlabeled mini-batches from the multiple domains $B^u = \{B_1^u, \cdots, B_M^u\}$.

5: L-Step: Calculate $\mathcal{L}_L = \mathcal{L}_c^1 + \mathcal{L}_c^2 + \alpha\mathcal{L}_\Delta$ on B^ℓ;
Update $\mathcal{F}_s$, $\{\mathcal{F}_d^m\}$, $\mathcal{C}_1$, $\mathcal{C}_2$ by descending along gradients $\Delta\mathcal{L}_L$.

6: A-Step: Calculate $\mathcal{L}_A = -(\mathcal{L}_c^1 + \mathcal{L}_c^2) + \mathcal{L}_{Adv}^u$ on B^ℓ and B^u, calculate $\mathcal{L}_{Adv}^d$ on B^ℓ and B^u;
Update $\mathcal{C}_1$ and $\mathcal{C}_2$ by ascending along $\Delta\mathcal{L}_A$;
Update $\mathcal{D}$ by ascending along $\Delta\mathcal{L}_{Adv}^d$.

7: R-Step: Calculate $\mathcal{L}_R = \mathcal{L}_{Adv}^u + \gamma\mathcal{L}_{Adv}^d$;
Update $\mathcal{F}_s$, $\{\mathcal{F}_d^m\}$ by descending along $\Delta\mathcal{L}_R$.

8: **end for**

4.3 Experiments

4.3.1 Implementation Details

In the experiments, we set the domain-invariant feature dimension as 128 and domain-specific feature dimension as 64, $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{D}$ are all MLPs whose input dimensions are 128+64, 128+64, and 128 respectively, they all have one hidden layer whose dimension is the same as the input dimension. We use ReLU as the activation function in each component. The dropout rate for each component is set as 0.4. The Adam optimizer [86] with a learning rate of 0.0001 is used to train the model. We set the batch size as 8 and the number of training iterations as 50. The proposed DACL model contains two hyperparameters, α and γ. we set $\alpha = 0.1$ and $\gamma = 0.1$. For the experiments on the Amazon review dataset, MLPs with two hidden layers are used as the feature extractors, the two hidden layers contain 1,000 and 500 units respectively. The input dimension of feature extractors is 5,000. For the experiments on the FDU-MTL dataset, CNN with one convolutional layer is used as the feature extractor. Its kernel sizes are (3, 4, 5), and the number of kernels is 200. The input dimension of the convolutional layer is 100, we use the word embedding, obtained by processing each word of the input sequence through word2vec [87], as the input of the feature extractors. For the evaluation, we use the average prediction probability scores of the two classifiers to make the final prediction of each instance.

4.3.2 Comparison baselines

We compare our DACL model with several state-of-the-art baselines:

- SVM, LR, MLP, BiLSTM, CNN: The support vector machine, logistic regression, multiple layer perceptron, Bi-directional long short-term memory, and convolutional neural network. For training these methods, we combine all training

data among multiple domains as the training set, ignoring the domain differences.

- MTLGraph: The Multi-task learning with graph regularization [88]. This method uses the sentiment word distribution similarity graph as its domain similarity graph.

- CMSC-LS, CMSC-SVM, CMSC-Log: The collaborative multi-domain sentiment classification method which trains a shareable classifier and a set of domain-specific classifiers for all domains. The final prediction is determined by the combination of the outputs of the shareable classifier and the corresponding classifier of the given instance. The CMSC model is trained with squared loss, hinge loss and, log loss respectively [85].

- MAN: The multinomial adversarial network for multi-domain text classification [10]. This method uses two forms of loss functions to train the multinomial domain discriminator: The negative log-likelihood loss (MAN-NLL) and the least square loss (MAN-L2). The MAN also induces the convergence conditional for adversarial MDTC with regard to the two loss functions.

- MT-CNN: The deep neural network model learns features across different tasks given very limited prior knowledge by using a single convolutional layer [58].

- MT-DNN: The multi-task deep neural network model with bag-of-words input and MLPs [89], in which a hidden layer is shared.

- ASP-MTL: The adversarial multi-task learning model for text classification [13], in which shared-private paradigm, adversarial training, and an orthogonal restriction are used to guide feature extraction.

All the comparison methods use the standard partitions of the datasets. We hence

Table 3: The classification accuracies of DACL on the Amazon review dataset.

Domain	SVM	LR	MLP	MTLGraph	CMSC-LS	CMSC-SVM	CMSC-Log	MAN-L2	MAN-NLL	DACL
Books	78.56	79.73	81.70	79.69	82.10	82.26	81.81	82.46	82.98	$\mathbf{83.45 \pm 0.14}$
DVD	78.66	80.14	81.65	81.84	82.40	83.48	83.73	83.98	84.03	$\mathbf{85.50 \pm 0.17}$
Electr.	83.03	84.54	85.45	83.69	86.12	86.76	86.67	87.22	87.06	$\mathbf{87.40 \pm 0.09}$
Kit.	84.74	86.10	85.95	87.06	87.56	88.20	88.23	88.53	88.57	$\mathbf{90.00 \pm 0.21}$
AVG	81.25	82.63	83.69	83.06	84.55	85.18	85.11	85.55	85.66	$\mathbf{86.59 \pm 0.10}$

take convenience to cite the results of SVM, LR, MLP, BiLSTM, CNN, MTLGraph, CMSC-LS, CMSC-SVM, CMSC-Log, MAN-NLL, MAN-L2 from [10], and the results of MT-CNN, MT-DNN, ASP-MTL from [19] for fair comparisons.

4.3.3 Experimental Results

We run the experiments in 5 trial runs and report the average classification accuracy with standard deviation. The experimental results on the Amazon review dataset and FDU-MTL dataset are reported in Table 3 and Table 4, respectively. The recent studies on the Amazon review dataset have shown that it is very difficult to make performance advances on this dataset. From Table 3, we observe that our dual adversarial co-learning mechanism enables the model to achieve the best performance on all domains of the Amazon review dataset compared with other baselines, including the ones that also adopt adversarial training (MAN-L2&MAN-NLL). In particular, our DACL approach can outperform the simple baselines (i.e., SVM, LR, and MLP) significantly, indicating that considering domain differences can contribute to the system performance in training MDTC models. Moreover, the performance gains obtained by our DACL method are more notable than the performance difference between any two baselines, this suggests that our dual adversarial co-learning mechanism is more effective.

For the experimental results on FDU-MTL, reported in Table 4, it can be noted that our proposed DACL approach outperforms BiLSTM, CNN, MT-CNN, and MT-DNN consistently on all domains with notable large performance gains. When compared with the recent baselines, like MAN-L2, MAN-NLL, and ASP-MTL, DACL obtains the best performance on 12 out of 16 domains and achieves the best average classification accuracy. The experimental results again validate the efficacy of our proposed DACL method.

4.3.4 Multi-Source Unsupervised Domain Adaptation

In many real-world applications, the availability of labeled data varies across different domains. It is hence important to evaluate the performance of the MDTC models on unseen domains. Multi-source unsupervised domain adaptation (MS-UDA) is a scenario where there exist multiple source domains with labeled data and one target domain with unlabeled data. Standard unsupervised domain adaptation transfers knowledge from one source domain to one target domain, MS-UDA allows adaptation from multiple source domains to a single target domain. As our proposed DACL method can leverage unlabeled data across different domains for feature representation extraction with a dual adversarial co-learning mechanism, it is supposed to be effective in MS-UDA. When evaluating the target data, as there is no labeled data in the target domain, we use the domain-invariant features as the inputs to the two classifiers $\mathcal{C}_1$ and $\mathcal{C}_2$, while the domain-specific features are set to be 0s.

The MS-UDA experiments are conducted on the Amazon review dataset, we follow the setting as shown in [10]. For each of the four separate experiments, three out of four domains are used as the source domains, while the remaining one is treated as the target domain. The evaluation is conducted on the target domain.

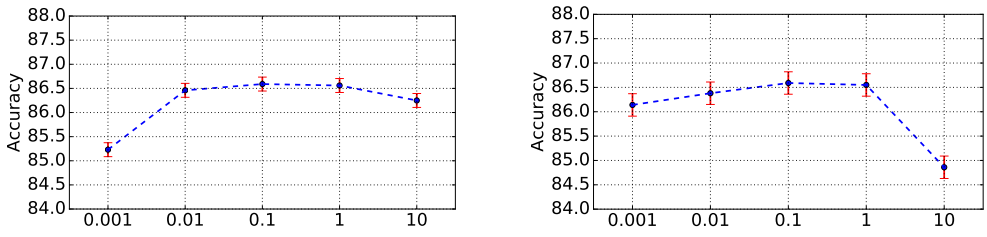
The baselines used in the MS-UDA task include: (1) A MLP trained on the source domains. (2) Two single-source domain adaptation methods, marginalized denoising

autoencoder (mSDA) [90] and domain adversarial neural network (DANN) [24]. When training mSDA and DANN, we combine the data from all source domains into a single source domain. (3) The state-of-the-art MS-UDA methods, multi-source domain adaptation neural networks (MDAN(H) and MADN(S)) [91] and MAN (MAN-L2 and MAN-NLL) [10]. The comparison results are reported in Table 5. We can see that the average results of most MS-UDA methods are better than that of the single-source domain adaptation methods. The proposed DACL model outperforms all the baselines in three out of the total four domains. In terms of average accuracy, the DACL method also outperforms other baselines. This suggests that our method has a good capacity to generalize learned knowledge to unseen domains.

4.3.5 Ablation Study

Since the proposed DACL model has five components, that are involved in two adversarial training mechanisms. In the ablation study, we investigate how each component impacts the performance of the DACL approach. In particular, we induce two ablation variants: (1) DACL w/o $\mathcal{D}$, the variant of the proposed DACL model without the discriminator $\mathcal{D}$, which hence drops the domain-based adversarial learning; (2) DACL w/o classifier $\mathcal{C}_2$, the variant of the proposed DACL model without the second classifier, which hence drops the discrepancy-based adversarial learning. The ablation study is conducted on the Amazon review dataset.

The experimental results between these two variants and the full model are reported in Table 6. We can see both variants induce inferior results, and the full model with dual adversarial training mechanisms produces the best results. This validates the contributions of both adversarial training mechanisms.



(a) α (b) γ

Figure 4: Parameter sensitivity analysis of the DACL

4.3.6 Parameter Sensitivity Analysis

The proposed DACL approach has two hyperparameters, α and γ. α is used to weight the separation regularizer; While γ is used to balance the two adversarial losses. We conduct the experiments on the Amazon review dataset to perform sensitivity analysis on these hyperparameters. We first fix $\gamma = 0.1$ and conduct experiments with different α values in the range $\{0.001, 0.01, 0.1, 1, 10\}$. Then we fix $\alpha = 0.1$ and conduct experiments with different γ values in the same range $\{0.001, 0.01, 0.1, 1, 10\}$. The experimental results are reported in Figure 4. The results are the average classification accuracy across multiple domains on the Amazon review dataset. From Figure 4, we observe that the average classification accuracy increases quickly with α value increasing from 0.001 to 0.1. This suggests that the separation regularizer is useful to encode different aspects of the input data. With α changing from 0.1 to 1, the performance change is very small, while the further increase of α degrades the performance. This suggests this regularizer can not be overemphasized, and a value in [0.01, 1] should be reasonable for α. For γ, the performance is robust for γ value with $(0.001, 1]$, but degrades when γ becomes large. This suggests that both adversarial training terms work.

4.4 Summary

In this chapter, we focus on the first and second challenges shown in Section 1.2: how to explicitly model the relationship among the multiple domains and how to further reduce the domain divergence among domains in the latent space. For this purpose, we present a dual adversarial co-learning (DACL) approach which leverages all available resources across different domains to advance MDTC. The model uses a shared feature extractor across different domains to capture domain-invariant features and a set of domain-specific feature extractors to capture domain-specific features. Two adversarial training mechanisms, which enforce multi-domain alignment through domain discriminator and improve labeled-unlabeled data alignment through maximal prediction discrepancy minimization, are incorporated to boost feature representation learning. Experimental results show that the proposed approach can improve the performance of MDTC effectively. In the next chapter, we will introduce mixup regularized adversarial networks for multi-domain text classification.

Table 4: The classification accuracies of DACL on the FDU-MTL dataset.

Domain	BiLSTM	CNN	MT-CNN	MT-DNN	ASP-MTL	MAN-L2	MAN-NLL	DACL
books	81.0	85.3	84.5	82.2	84.0	**87.6**	86.8	87.5 ± 1.0
electronics	78.5	87.8	83.2	81.7	86.8	87.4	88.8	$\mathbf{90.3 \pm 0.2}$
dvd	80.5	76.3	84.0	84.2	85.5	88.1	88.6	$\mathbf{89.8 \pm 0.3}$
kitchen	81.2	84.5	83.2	80.7	86.2	89.8	89.9	$\mathbf{91.5 \pm 0.7}$
apparel	86.0	86.3	83.7	85.0	87.0	87.6	87.6	$\mathbf{89.5 \pm 0.4}$
camera	86.0	89.0	86.0	86.2	89.2	91.4	90.7	$\mathbf{91.5 \pm 0.3}$
health	78.7	87.5	87.2	85.7	88.2	89.8	89.4	$\mathbf{90.5 \pm 0.3}$
music	77.2	81.5	83.7	84.7	82.5	85.9	85.5	$\mathbf{86.3 \pm 0.2}$
toys	84.7	87.0	89.2	87.7	88.0	90.0	90.4	$\mathbf{91.3 \pm 0.2}$
video	83.7	82.3	81.5	85.0	84.5	89.5	**89.6**	88.5 ± 0.1
baby	83.5	82.5	87.7	88.0	88.2	90.0	90.2	$\mathbf{92.0 \pm 0.5}$
magazine	91.5	86.8	87.7	89.5	92.2	92.5	92.9	$\mathbf{93.8 \pm 0.1}$
software	85.7	87.5	86.5	85.7	87.2	90.4	**90.9**	90.5 ± 0.2
sports	84.0	85.3	84.0	83.2	85.7	89.0	89.0	$\mathbf{89.3 \pm 0.5}$
IMDb	85.0	83.3	86.2	83.2	85.5	86.6	87.0	$\mathbf{87.3 \pm 0.1}$
MR	74.7	75.5	74.5	75.5	**76.7**	76.1	**76.7**	76.0 ± 0.3
AVG	82.6	84.3	84.5	84.3	86.1	88.2	88.4	$\mathbf{89.1 \pm 0.2}$

Table 5: MS-UDA classification accuracies of DACL on the Amazon review dataset

Domain	MLP	mSDA	DANN	MDAN(H)	MDAN(S)	MAN-L2	MAN-NLL	DACL
Books	76.55	76.98	77.89	78.45	78.63	78.45	77.78	**80.22**
DVD	75.88	78.61	78.86	77.97	80.65	81.57	82.74	**82.96**
Elec.	84.60	81.98	84.91	84.83	**85.34**	83.37	83.75	84.90
Kit.	85.45	84.26	86.39	85.80	86.26	85.57	86.41	**86.75**
AVG	80.46	80.46	82.01	81.76	82.72	82.24	82.67	**83.71**

Table 6: Ablation study analysis of the DACL.

Method	Books	DVD	Electr.	Kit.	AVG
DACL (full)	83.45	85.50	87.40	90.00	86.59
DACL w/o $\mathcal{D}$	82.90	85.10	86.30	89.15	85.86
DACL w/o $\mathcal{C}_2$	81.85	83.10	85.75	89.10	84.95

Chapter 5

Mixup Regularized Adversarial Networks for Multi-Domain Text Classification

5.1 Introduction

As mentioned in Section 1.1, multi-domain text classification (MDTC) can not only train models on labeled text data but also consider relatedness across multiple domains. MDTC seeks to simultaneously utilize all available data across multiple domains to improve the classification accuracy of each domain. Recently, DNN-based models have been widely used for multi-domain learning in both computer vision [83, 84] and NLP [10, 85]. Most recent state-of-the-art MDTC methods adopt the shared-private paradigm [18] and adversarial training [21] to improve the system performance. The adversarial training is used to align multiple feature distributions in the shared latent space to generate domain-invariant features [10]. The shared-private paradigm captures both domain-invariant features and domain-specific features and enables these two forms of features to complement each other [19]. The idea behind integrating adversarial training and shared-private paradigm with MDTC is to perform domain alignment across domains and guarantee the optimum separations among the shared latent space and multiple domain-specific feature spaces, preventing

domain-specific features from emerging into the shared latent space. However, these approaches still face two challenges: (1) the mini-batch stochastic gradient descent is often used in neural network training, if the batch size is small, the training data drawn in each iteration may not be sufficient to guarantee strong domain-invariance of the shared latent space; (2) lacking sufficient labeled data in each domain can not guarantee the strong discriminability of the learned features [92].

To address the aforementioned problem, in this chapter, we propose mixup regularized adversarial networks (MRANs), which introduce two types of mixup-based interpolation regularizers to enrich the feature patterns in the latent space. Mixup [93] is a form of vicinal risk minimization [94], which trains models on convex combinations of training instances and labels to overcome disadvantages of empirical risk minimization, such as memorization of corrupted labels and instability of adversarial training. Inspired by mixup, this chapter introduces the domain mixup regularizer and the category mixup regularizer. In particular, the domain mixup regularizer seeks to expand the searching range in the shared latent space, excavating more intrinsic feature patterns such that a more continuous shared latent space can be achieved. The category mixup regularizer aims to enforce consistent predictions on the linear interpolations for both labeled and unlabeled instances, imposing stronger discriminability to the learned features. With the enhancement of these two regularizers, our proposed MRAN approach can avoid mismatching features with different labels and generalize well across multiple domains. Empirical analyses are carried out on two MDTC benchmarks: the Amazon review dataset and the FDU-MTL dataset. The experimental results show that our MRAN method outperforms state-of-the-art baselines on both datasets, validating the efficacy of the MRAN method.

- We propose a mixup regularized adversarial network for multi-domain text classification, which introduces two mixup-based regularizers: the domain mixup regularizer, which excavates more intrinsic feature patterns in the shared latent

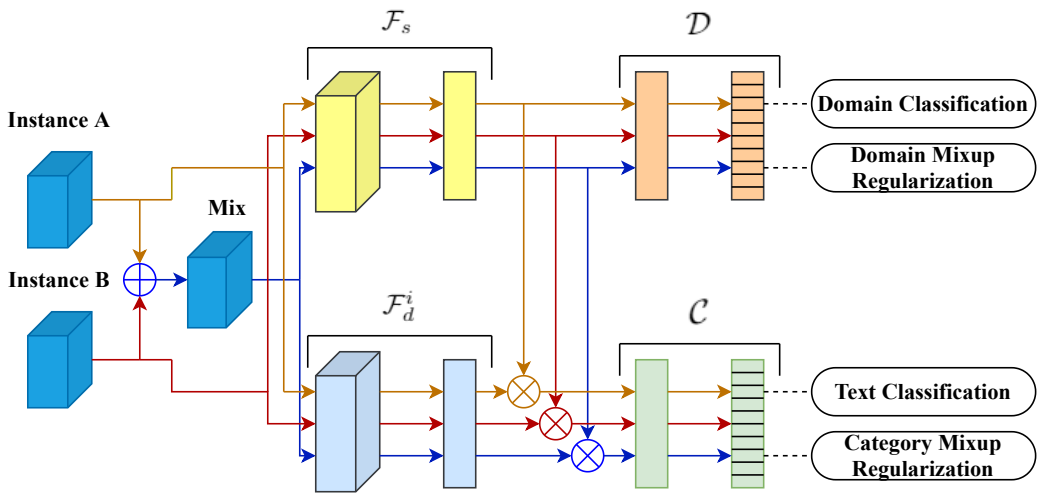


Figure 5: The architecture of the MRAN.

space, and the category mixup regularizer, which enforces consistent predictions on the linear interpolations for both labeled and unlabeled instances.

- We evaluate our MRAN approach on two MDTC benchmarks to show that the performance of the MRAN can outperform the state-of-the-art baselines.

- We conduct extensive analyses to show how each component contributes to the system performance and how each hyperparameter influences the system performance.

5.2 Mixup Regularized Adversarial Learning

As illustrated in Figure 5, our proposed Mixup Regularized Adversarial Network (MRAN) is based on the shared-private paradigm. There are four components in MRAN: a shared feature extractor $\mathcal{F}_s$ is used to generate domain-invariant features, M domain-specific feature extractors $\{\mathcal{F}_d^i\}_{i=1}^M$ are used to generate domain-specific features, a multinomial domain discriminator $\mathcal{D}$ is used to distinguish features among

multiple domains, and a classifier $\mathcal{C}$ is used to conduct classification. We use domain-invariant features as the input of the domain discriminator $\mathcal{D}$, while the concatenations of domain-invariant features and domain-specific features are used as the input of the classifier $\mathcal{C}$. The main goal of MRAN is to leverage all available resources to improve the average classification accuracy across all domains. In order to achieve this goal, (1) the shared feature space should contain sufficient feature patterns such that the discriminability of the learned features can be guaranteed; (2) As domain-specific features only contribute to their corresponding domains, they are not encouraged to emerge into the shared latent space, which will lead to performance degradation.

5.2.1 Multinomial Adversarial Learning

In the proposed MRAN model, input data from each domain has two data flows: one is to extract the domain-invariant features which can enable knowledge sharing across all domains, while the other one is to capture domain-specific features which can maximally represent useful information of each individual domain. These two flows are supposed to complement each other. The multinomial domain discriminator $\mathcal{D}$ takes a shared feature vector as its input and outputs the likelihood of that instance coming from each domain; its main aim is to discriminate training instances from different domains. For M domains, $\mathcal{D}$ will be a M-class classifier and we use $\mathcal{D}_i(\mathcal{F}_s(\mathbf{x}))$ to denote the predicted probability of instance $\mathbf{x}$ coming from the i-th domain. A two-player minimax game is played between the multinomial domain discriminator $\mathcal{D}$ and the shared feature extractor $\mathcal{F}_s$, where the discriminator $\mathcal{D}$'s goal is trying to distinguish features among different domains, while the shared feature extractor $\mathcal{F}_s$'s goal is to maximally confuse the discriminator such that the features from different domains are indistinguishable in the shared latent space. We use the negative log-likelihood loss (NLL) to encode the multinomial adversarial loss function as follows:

$$\mathcal{L}_{Adv} = -\sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i \sim \mathbb{L}_i \cup \mathbb{U}_i} \log[\mathcal{D}_i(\mathcal{F}_s(\mathbf{x}_i))] \tag{5.1}$$

This formulation extends the standard binary adversarial learning to multi-class adversarial learning which suits the MDTC scenario. After this learning, the shared feature extractor $\mathcal{F}_s$ and the discriminator $\mathcal{D}$ reach an equilibrium at which these two components both cannot improve such that the discriminator $\mathcal{D}$ is unable to distinguish features among multiple domains.

We also adopt the NLL to encode the classification loss. The classifier $\mathcal{C}$ utilizes multiple-layer perceptrons (MLPs) with a softmax layer on top and takes the concatenation of shared and domain-specific features and outputs the probability vector of the given instance. To impose discriminability to the learned features, the classifier $\mathcal{C}$ is trained together with the feature extractors by minimizing the NLL loss on the labeled data:

$$\mathcal{L}_c = -\sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathbb{L}_i} \log[\mathcal{C}_y(\mathcal{F}^i(\mathbf{x}_i)] \tag{5.2}$$

Where $\mathcal{F}^i(\mathbf{x})$ is the concatenation of $\mathcal{F}_s(\mathbf{x})$ and $\mathcal{F}_d^i(\mathbf{x})$, $\mathcal{C}_y(\mathcal{F}^i(\mathbf{x}))$ denotes the probability of instance $\mathbf{x}$ belonging to the category y.

5.2.2 Mixup-based Regularization

The limited amounts of labeled data in each domain could lead to overfitting or even accuracy deterioration. In order to address this issue, we introduce an interpolation regularization based on the mixup data augmentation technique. Mixup [93] constructs virtual instances using convex combinations of a pair of instances $(\mathbf{x}^k, \mathbf{x}^s)$, and implements the same linear interpolation on their corresponding labels (y^k, y^s) to generate pseudo-labels.

$$\widetilde{\mathbf{x}} = \mathcal{M}_\lambda(\mathbf{x}^k, \mathbf{x}^s) = \lambda \mathbf{x}^k + (1-\lambda)\mathbf{x}^s \tag{5.3}$$

$$\widetilde{y} = \mathcal{M}_\lambda(y^k, y^s) = \lambda y^k + (1-\lambda) y^s \tag{5.4}$$

Where $\lambda \in [0, 1]$ is the mixup ratio, drawn from $Beta(\alpha, \alpha)$ distribution. In all our experiments, α is set as 0.2. Mixup has been proven effective in both the input space and the latent space to improve performance on various tasks [93, 95]. In MRAN, we develop category and domain mixup regularizers on the instance level.

The category mixup regularizer involves two parts associated with labeled and unlabeled data. we first apply mixup on the labeled instances in each domain to enforce prediction consistency:

$$\mathcal{L}_a^M = -\sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x}_i^k, y_i^k), (\mathbf{x}_i^s, y_i^s) \sim \mathbb{L}_i} \log[\mathcal{C}_{\widetilde{y}_i}(\mathcal{F}_i(\widetilde{\mathbf{x}}_i)] \tag{5.5}$$

Then, for the unlabeled instances, as we have no direct access to the label information, we use pseudo-labels to replace real labels, and mixup is applied to the instances and the corresponding pseudo-labels. We assume that the model should make predictions linearly in-between instances. Thus, we introduce a penalty term on the unlabeled data which enforces consistent regularization between the prediction of interpolation of samples and the interpolation of the predictions of these samples:

$$\mathcal{L}_u^M = \sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x}_i^k, \mathbf{x}_i^s) \sim \mathbb{U}_i} D(p(\widetilde{x}_i), \mathcal{M}_\lambda(p(\mathbf{x}_i^k), p(\mathbf{x}_i^s))) \tag{5.6}$$

Where $p(\mathbf{x}) = \log[\mathcal{C}(\mathcal{F}^i(\mathbf{x}))]$, $\mathcal{C}(\cdot)$ denotes the softmax prediction of the classifier $\mathcal{C}$, $D(\cdot, \cdot)$ denotes the discrepancy metric. In all our experiments, we use the ℓ_1 norm function. The category mixup regularizer can explore more intrinsic feature patterns of the shared latent space, enhancing the discriminability of the domain-invariant features.

For the domain mixup regularizer, we conduct mixup interpolation on instances for each domain. Since the domain label of the instances sampled from a specific domain should be identical, it is not necessary to implement Mixup on their domain labels. The domain mixup regularizer is defined as follows:

$$\mathcal{L}_{Adv}^{M} = -\sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x}_i^k, \mathbf{x}_i^s)\sim \mathbb{L}_i \cup \mathbb{U}_i} log[\mathcal{D}_i(\mathcal{F}_s(\widetilde{\mathbf{x}}_i))] \tag{5.7}$$

With the domain mixup regularizer, a more continuous shared latent space can be achieved such that the generalization of the learned features can be boosted.

5.2.3 Training Procedure

The final loss function of our proposed MRAN is defined as follows:

$$\min_{\mathcal{F}_s, \{\mathcal{F}_d^i\}, C} \max_{\mathcal{D}} \quad \mathcal{L}_c + \lambda_d \mathcal{L}_{Adv} + \lambda_a \mathcal{L}_a^M + \lambda_u \mathcal{L}_u^M + \lambda_m \mathcal{L}_{Adv}^M \tag{5.8}$$

The MRAN training process is illustrated in Algorithm 2. The model is trained with a mini-batch stochastic gradient descent algorithm. In each iteration, the samples are first mixed in the input space to generate the virtual samples and the corresponding labels per domain. Specifically, a labeled instance is mixed with another labeled instance, while an unlabeled instance is mixed with another unlabeled instance. The mixed samples are supposed to preserve their inputs' category and domain information. Then, the mixed samples are used in the regularization terms. The category mixup regularizer can enforce the consistent prediction constraint, while the domain mixup regularizer enriches the intrinsic feature patterns in the latent space such that the latent space is expected to be more domain-invariant.

Algorithm 2 Stochastic gradient descent training algorithm

1: **Input:** labeled data $\mathbb{L}_i$ and unlabeled data $\mathbb{U}_i$ in M domains; hyperparameters α ,λ_d, λ_a, λ_u,λ_m

2: **for** number of training iterations **do**

3: Sample labeled mini-batches from the multiple domains $B^{\ell} = \{B_1^{\ell}, \cdots, B_M^{\ell}\}$.

4: Sample unlabeled mini-batches from the multiple domains $B^u = \{B_1^u, \cdots, B_M^u\}$.

5: Calculate $l_D = \lambda_d \mathcal{L}_{Adv} + \lambda_m \mathcal{L}_{Adv}^M$ on B^{ℓ} and B^u;
Update $\mathcal{D}$ by ascending along gradients ∇l_D.

6: Calculate $loss = \mathcal{L}_c + \lambda_a \mathcal{L}_a^M + \lambda_u \mathcal{L}_u^M + \lambda_d \mathcal{L}_{Adv} + \lambda_m \mathcal{L}_{Adv}^M$ on B^{ℓ} and B^u;
Update $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}$, $\mathcal{C}$ by descending along gradients $\nabla loss$.

7: **end for**

5.3 Experiments

5.3.1 Comparison Methods

We compare our MRAN model with several state-of-the-art baselines:

- CMSC-LS, CMSC-SVM, CMSC-Log: The collaborative multi-domain sentiment classification method which trains a shareable classifier and a set of domain-specific classifiers for all domains. The final prediction is determined by the combination of the outputs of the shareable classifier and the corresponding classifier of the given instance. The CMSC model is trained with squared loss, hinge loss, and log loss respectively [85].

- MAN: The multinomial adversarial network for multi-domain text classification [10]. This method uses two forms of loss functions to train the multinomial domain discriminator: The negative log-likelihood loss (MAN-NLL) and the least square loss (MAN-L2). The MAN also induces the convergence conditional

for adversarial MDTC with regard to the two loss functions.

- MT-CNN: The deep neural network model learns features across different tasks given very limited prior knowledge by using a single convolutional layer [58].
- MT-DNN: The multi-task deep neural network model with bag-of-words input and MLPs [89], in which a hidden layer is shared.
- ASP-MTL: The adversarial multi-task learning model for text classification [13], in which shared-private paradigm, adversarial training, and an orthogonal restriction are used to guide feature extraction.
- MT-BL: The multi-task learning with bidirectional language [76], which introduces language modeling as an auxiliary task to encourage the domain-specific feature extractors to capture more syntactic and semantic information, and a uniform label distribution-based loss constraint to the shared feature extractor to enhance the ability to learn domain-invariant features.
- DACL: The dual adversarial co-learning method [16], which combines the discriminator-based adversarial training and classifier-based adversarial training to align not only the multiple feature distributions but also the labeled and unlabeled features in the latent space.

All the comparison methods use the standard partitions of the datasets. Thus, we cite the results from [10, 19, 76] for fair comparisons.

5.3.2 Implementation Details

We implemented our approach using Pytorch. The proposed MRAN has four hyper-parameters, λ_d, λ_a, λ_u and λ_m. In the experiments, we fixed $\lambda_d = 1$, $\lambda_a = 0.001$, $\lambda_u = 0.1$ and $\lambda_m = 0.00001$. In the experiments, we set the domain-invariant feature

dimension as 128 and domain-specific feature dimension as 64, $\mathcal{C}$ and $\mathcal{D}$ are both MLPs whose input dimensions are 128+64 and 128 respectively, they both have one hidden layer whose dimension is the same as the input dimension. We use ReLU as the activation function in each component. The dropout rate for each component is set as 0.4. The Adam optimizer [86] with a learning rate of 0.0001 is used to train the model. We set the batch size as 8 and the number of training iterations as 50. For the experiments on the Amazon review dataset, MLPs with two hidden layers are used as the feature extractors, the two hidden layers contain 1,000 and 500 units respectively. The input dimension of feature extractors is 5,000. For the experiments on the FDU-MTL dataset, CNN with one convolutional layer is used as the feature extractor. Its kernel sizes are (3, 4, 5), and the number of kernels is 200. The input dimension of the convolutional layer is 100, we use the word embedding, obtained by processing each word of the input sequence through word2vec [87], as the input of the feature extractors.

5.3.3 Results

We run the experiments in 5 trial runs and report the average classification accuracy with standard deviation. The experimental results on the Amazon review dataset and the FDU-MTL dataset are shown in Table 3 and Table 4, respectively. From Table 3, we can see that our model achieves the highest average accuracy of 87.64%, surpassing MAN-NLL by a significant margin of 1.98%. Moreover, our model performs best on all the domains compared with the other methods.

From the experimental results on the FDU-MTL dataset, reported in Table 4. We can see that, in terms of average accuracy, our MRAN can produce competitive performance compared with state-of-the-art baselines. Moreover, our model performs best on 7 of the 16 domains. The highest accuracy for the domains books, health, toys, and software is obtained by MT-BL. However, MT-BL adds language modeling

Table 3: The classification accuracies of MRAN on the Amazon review dataset.

Domain	CMSC-LS	CMSC-SVM	CMSC-Log	MAN-L2	MAN-NLL	DACL	MRAN
Books	82.10	82.26	81.81	82.46	82.98	83.45	$\mathbf{84.60 \pm 0.18}$
DVD	82.40	83.48	83.73	83.98	84.03	85.50	$\mathbf{85.60 \pm 0.16}$
Electr.	86.12	86.76	86.67	87.22	87.06	87.40	$\mathbf{89.10 \pm 0.05}$
Kit.	87.56	88.20	88.23	88.53	88.57	90.00	$\mathbf{91.25 \pm 0.19}$
AVG	84.55	85.18	85.11	85.55	85.66	86.59	$\mathbf{87.64 \pm 0.08}$

as an auxiliary task for domain-specific feature extractor learning by using another Bidirectional long short-term memory network (BiLSTM), which makes the model architecture more complicated. The experimental results on the FDU-MTL dataset again illustrate the effectiveness of our model on the MDTC task.

In summary, different from other methods which adopt the shared-private paradigm, our MRAN uses mixup to conduct interpolation regularization on both category and domain levels to help enrich the intrinsic feature patterns in the shared latent space, leading to stronger discriminability and transferability of the learned features.

5.3.4 Multi-Source Unsupervised Domain Adaptation

When tackling MDTC tasks, the model requires labeled data for each domain. While in real-world applications, many domains may not have any labeled data. Therefore, it is necessary to explore the generalization ability of our method when applied to an unseen domain without labeled data.

Standard unsupervised domain adaptation method transfers knowledge from one source domain, which has abundant labeled data, to one target domain, which has no labeled data. Multi-source unsupervised domain adaptation (MS-UDA) allows

adaptation from multiple source domains to a single target domain. In the MS-UDA setting, we have multiple source domains with both labeled and unlabeled data and one target domain with unlabeled data. Since the MRAN can leverage unlabeled data across different domains to do representation learning, it is supposed to be effective in the MS-UDA setting. Since the target domain has no labeled data, we only feed the shared features to $\mathcal{C}$, while the domain-specific part is set to 0.

The baselines used in the MS-UDA task include: (1) A MLP trained on the source domains. (2) Two single-source domain adaptation methods, marginalized denoising autoencoder (mSDA) [90] and domain adversarial neural network (DANN) [24]. When training mSDA and DANN, we combine the data from all source domains into a single source domain. (3) The state-of-the-art MS-UDA methods, multi-source domain adaptation neural networks (MDAN(H) and MADN(S)) [91] and MAN (MAN-L2 and MAN-NLL) [10].

We conduct the MS-UDA experiments on the Amazon review dataset, following the same setting as [10]. For each of the four separate experiments, three of the four domains are treated as the source domains, and the remaining one is used as the target domain. The test results are evaluated on the target domain. In Table 5, the target domain is shown in the left column, and the test set accuracy is presented for various methods. We can see that our MRAN performs better than the baselines (MLP, mSDA, and DANN) which are developed on a single source domain composed of all the training data from multiple source domains. When compared with MDAN and MAN, our method can also produce superior results not only in terms of average accuracy, but also for the accuracy of each individual domain, which demonstrates that our model has a good ability for generalization.

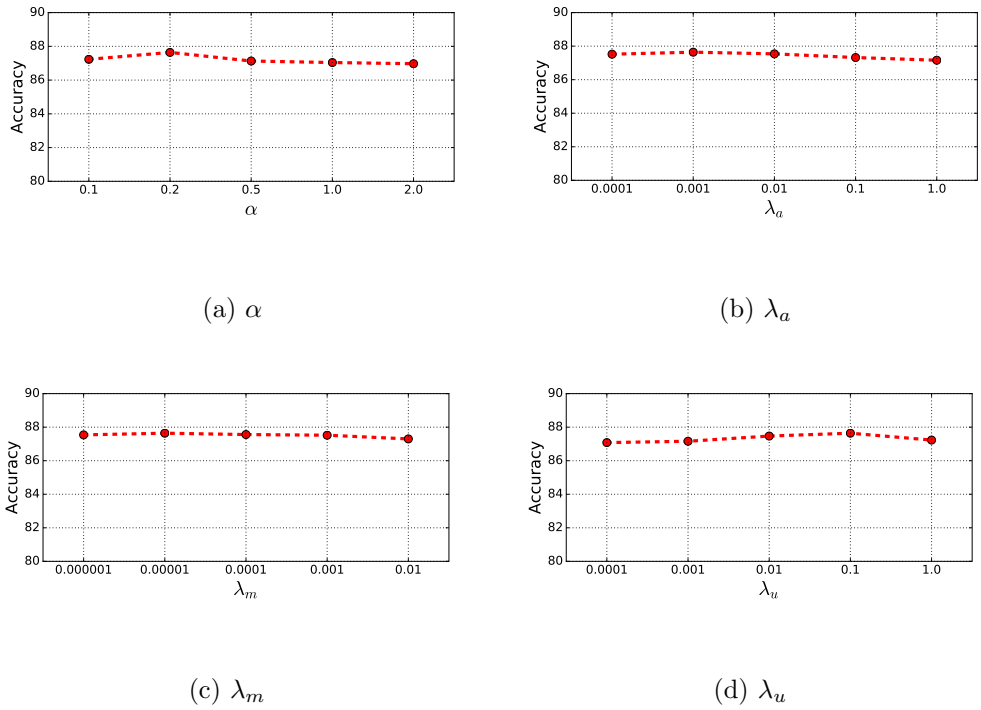
5.3.5 Ablation Study

In order to verify the effectiveness of domain mixup (DM) regularizer and category mixup (CM) regularizer, we conduct an extensive ablation study on the Amazon review dataset. In addition, for the category mixup regularizer, we both evaluate the contributions of the category mixup regularizer on labeled data (LCM) and the category mixup regularizer on unlabeled data (UCM). From Table 6, we observe that all the four variants obtain inferior results, and the full MRAN produces the best results. The model without CM performs the worst, which shows that CM can contribute more to the performance than DM. Moreover, both LCM and UCM, which help boost the discriminability of the domain-invariant features, contribute to the improved performance of our model.

5.3.6 Parameter Sensitivity Analysis

In this section, we explore the sensitivity of our method to the values of hyperparameters α, λ_a, λ_u and λ_m. α controls the selection of λ which influences the virtual sample generation, λ_a, λ_u and λ_m are used to balance among different losses. We evaluate these hyperparameters on the Amazon review dataset in terms of the average classification accuracy. When evaluating one hyperparamter, the others are fixed to their default values (e.g., $\alpha = 0.1$, $\lambda_a = 0.001$, $\lambda_u = 0.1$ and $\lambda_m = 0.00001$). α is tested in the range 0.1, 0.2, 0.5, 1.0, 2.0, λ_a and λ_u are investigated in the range 0.0001, 0.001, 0.01, 0.1, 1.0, and λ_m is evaluated in the range 0.000001, 0.00001, 0.0001, 0.001, 0.01. The experimental results are shown in Figure 5.

From Figure 6, we can see that all four hyperparameters α, λ_a, λ_u and λ_m can not significantly influence the average classification accuracy. In consequence, we can set them to their default values for all experiments, and the classification accuracy should be satisfactory.



(a) α (b) λ_a

(c) λ_m (d) λ_u

Figure 6: Parameter sensitivity analysis of MRAN

5.4 Summary

In this chapter, we aim to address the third MDTC challenge shown in Section 1.2: how to use limited amounts of data from each domain to guarantee the domain-invariance of the shared latent space. We propose the mixup regularized adversarial networks (MRANs) which introduce the category mixup regularizer to enforce consistent predictions in-between training instances, imposing stronger discriminability to the learned features, and the domain mixup regularizer to explore more internal features in the shared latent space, leading to a more continuous shared latent space. These two mixup-based interpolation regularizers can enhance and complement each other. The experiments on two MDTC benchmarks show that our MRAN can effectively improve system performance. In the next chapter, we will propose a conditional adversarial learning mechanism for multi-domain text classification and provides its

theoretical guarantee.

Table 4: The classification accuracies of MRAN on the FDU-MTL dataset.

Domain	MT-CNN	MT-DNN	ASP-MTL	MAN-L2	MAN-NLL	MT-BL	DACL	MRAN
books	84.5	82.2	84.0	87.6	86.8	**89.0**	87.5	87.0 ± 0.2
electronics	83.2	81.7	86.8	87.4	88.8	90.2	**90.3**	89.0 ± 0.7
dvd	84.0	84.2	85.5	88.1	88.6	88.0	**89.8**	89.0 ± 0.4
kitchen	83.2	80.7	86.2	89.8	89.9	90.5	91.5	$\mathbf{93.0 \pm 0.2}$
apparel	83.7	85.0	87.0	87.6	87.6	87.2	89.5	$\mathbf{91.5 \pm 0.3}$
camera	86.0	86.2	89.2	91.4	90.7	89.5	91.5	$\mathbf{93.0 \pm 0.4}$
health	87.2	85.7	88.2	89.8	89.4	**92.5**	90.5	90.0 ± 0.3
music	83.7	84.7	82.5	85.9	85.5	86.0	86.3	$\mathbf{86.5 \pm 0.1}$
toys	89.2	87.7	88.0	90.0	90.4	**92.0**	91.3	86.0 ± 0.3
video	81.5	85.0	84.5	89.5	**89.6**	88.0	88.5	88.5 ± 0.4
baby	87.7	88.0	88.2	90.0	90.2	88.7	**92.0**	90.0 ± 0.5
magazine	87.7	89.5	92.2	92.5	92.9	92.5	**93.8**	93.5 ± 0.3
software	86.5	85.7	87.2	90.4	90.9	**91.7**	90.5	89.5 ± 0.4
sports	84.0	83.2	85.7	89.0	89.0	89.5	89.3	$\mathbf{90.5 \pm 0.2}$
IMDb	86.2	83.2	85.5	86.6	87.0	88.0	87.3	$\mathbf{89.0 \pm 0.1}$
MR	74.5	75.5	76.7	76.1	76.7	75.7	76.0	$\mathbf{78.5 \pm 0.5}$
AVG	84.5	84.3	86.1	88.2	88.4	88.6	**89.1**	89.0 ± 0.1

Table 5: Multi-source unsupervised domain adaptation results of MRAN on the Amazon review dataset.

Domain	MLP	mSDA	DANN	MDAN(H)	MDAN(S)	MAN-L2	MAN-NLL	MRAN
Books	76.55	76.98	77.89	78.45	78.63	78.45	77.78	**80.92**
DVD	75.88	78.61	78.86	77.97	80.65	81.57	82.74	**83.71**
Elec.	84.60	81.98	84.91	84.83	85.34	83.37	83.75	**85.46**
Kit.	85.45	84.26	86.39	85.80	86.26	85.57	86.41	**87.08**
AVG	80.46	80.46	82.01	81.76	82.72	82.24	82.67	**84.11**

Table 6: Ablation study analysis of MRAN.

Method	Books	DVD	Electr.	Kit.	AVG
MRAN (full)	84.60	85.60	89.10	91.25	87.64
MRAN w/o DM	84.05	85.35	87.50	90.50	86.85
MRAN w/o CM	83.85	84.90	88.00	89.20	86.49
MRAN w/o LCM	83.60	85.00	87.20	90.25	86.51
MRAN w/o UCM	84.10	85.55	86.70	89.80	86.54

Chapter 6

Conditional Adversarial Networks for Multi-Domain Text Classification

6.1 Introduction

As mentioned in Section 1.1, multi-domain text classification (MDTC) can handle the scenario where labeled data exist for multiple domains, but in insufficient amounts to train an effective classifier. The most recent MDTC methods often adopt shared-private paradigm and adversarial training. The shared-private paradigm divides the latent space into two types: (1) the shared latent space with the aim of learning domain-invariant features; (2) the domain-specific latent spaces, each of which captures domain-specific features for the corresponding domain. The adversarial training, which is first proposed by generative adversarial networks [21], is used to explicitly ensure the optimum separations among the share latent space and multiple domain-specific latent spaces, preventing the domain-specific features from emerging into the shared latent space [19]. The adversarial training is also used to align multiple feature distributions in the shared latent space to generate domain-invariant features [10]. By integrating the adversarial training and shared-private paradigm into MDTC, the transferability and discriminability of the learned features can be boosted such that

these methods can yield state-of-the-art results [10, 16, 19]. However, these methods still face one major challenge: the data distributions are supposed to contain complex structures, when performing adversarial training, the alignment may risk matching features with different labels together, leading to weak discriminability of the learned features.

To address the aforementioned challenge, in this chapter, we propose conditional adversarial networks (CANs) for MDTC which optimizes the adversarial training through aligning joint distributions of domain-invariant features and label prediction. This method is inspired by the conditional generative adversarial networks (CGANs) [96], which aligns distributions of real and generated images via conditioning the generator and discriminator on extra information. The proposed CAN approach seeks to model domain variance in domain-invariant features and pseudo-labels to excavate the relationship between domain-invariant features and class-aware information conveyed by the label predictions. The ultimate goal of CAN is to impose stronger discriminability on the learned features such that the system performance can be improved. In addition, we also incorporate the entropy conditioning mechanism to avoid conditioning on the class-aware information with low certainty by giving easy-to-transfer instances higher priority in the adversarial training. The theoretical guarantee to support the CAN approach is also provided. The efficacy of our proposed CAN approach is validated on two MDTC benchmarks: the Amazon review dataset and the FDU-MTL dataset. The experimental results show that the CAN method yields competitive performance compared with the state-of-the-art baselines. The extensive experiments on multi-source unsupervised domain adaptation (MS-UDA) are carried out to demonstrate that our CAN method has the ability to generalize to unseen domains. The contribution of this chapter is summarized as follows:

- We propose a conditional adversarial network (CAN) for multi-domain text classification which incorporates conditional domain discriminator and entropy

conditioning to perform alignment on the joint distributions of shared features and label predictions to improve the system performance.

- We provide the theoretical guarantee of the CAN framework, demonstrating that CANs are minimizers of divergences among multiple joint distributions of shared features and label predictions, and providing the condition where the conditional domain discriminator reaches its optimum.

- We evaluate the effectiveness of CAN on two MDTC benchmarks. The experimental results show that the CAN method can beat other state-of-the-art baselines. In addition, the MS-UDA experiments illustrate that the CAN model can generalize in cases where no labeled data exist for a subset of domains.

6.2 Conditional Adversarial Learning

As shown in Figure 7, our proposed conditional adversarial network (CAN) consists of four components: a shared feature extractor $\mathcal{F}_s$, a set of domain-specific feature extractors $\{\mathcal{F}_d^i\}_{i=1}^M$, a conditional domain discriminator $\mathcal{D}$, and a text classifier $\mathcal{C}$. The shared feature extractor $\mathcal{F}_s$ aims to capture domain-invariant features shared across different domains. The domain-specific feature extractor $\mathcal{F}_d^i$ seeks to learn knowledge that is unique to its corresponding domain. The form of the feature extractor is flexible, it can adopt multiple layer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), or transformer. The feature extractor generates vectors with a fixed length, which is considered as the hidden representation of certain inputs. The conditional domain discriminator $\mathcal{D}$ is used to distinguish features among multiple domains. It takes the concatenation of a domain-invariant feature vector and a prediction vector of the given instance provided by the classifier $\mathcal{C}$ as input and outputs the probability of that instance coming from each domain. The

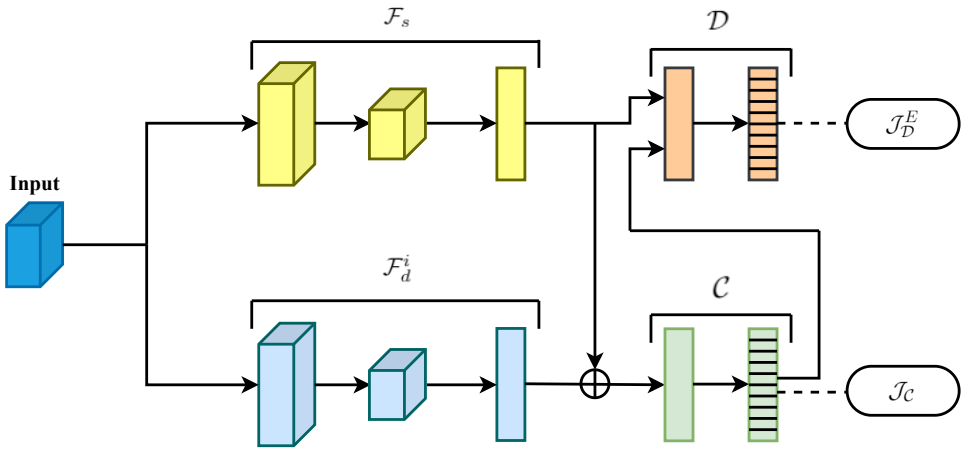


Figure 7: The architecture of the CAN model.

classifier $\mathcal{C}$ is used to conduct classification. It takes the concatenation of a domain-invariant feature vector and a domain-specific feature vector as input and outputs label probabilities.

In past decades, adversarial training has been widely investigated in various applications, such as image generation, adversarial attack, and domain alignment [10, 23, 45]. In domain alignment, it is used to minimize divergence across domains through a minimax optimization [91]. In the optimization, a domain discriminator and a feature extractor are trained against each other to generate the domain-invariant features. The loss function of the domain discriminator corresponds well to the domain divergence. Most recent MDTC methods adopt adversarial training, which aligns marginal distributions, yield state-of-the-art performance [10, 19]. However, the transferability with representation transition from general to specific along deep networks is decreasing significantly [97], aligning marginal distributions can not guarantee the global alignment among multiple domains. Moreover, the feature distributions in the shared latent space are supposed to embody complex structures, aligning marginal distributions may risk matching features with different labels.

The conditional generative adversarial network (CGAN) reveals that better alignment on two different distributions can be achieved by conditioning the generator and discriminator on class-aware information [96]. The discriminative information conveyed by the pseudo-labels potentially discloses the structure information underlying the feature distributions. Thus, conditional adversarial training has the ability to better model the domain divergence. Our proposed CAN method updates the domain discriminator such that the joint distributions of domain-invariant features and label predictions can be aligned. Our proposed CAN approach is trained in an alternating fashion: The shared feature extractor, domain-specific feature extractors, and the classifier are trained with the main optimizer, while the conditional domain discriminator is trained with a separate optimizer. We use $\mathcal{L}_{\mathcal{C}}$ and $\mathcal{L}_{\mathcal{D}}$ to denote the classification loss and the conditional adversarial loss, respectively. The negative log-likelihood (NLL) loss is used to encode these two loss functions:

$$\mathcal{L}_{\mathcal{C}}(\widetilde{y}, y) = -\log P(\widetilde{y} = y) \tag{6.1}$$

$$\mathcal{L}_{\mathcal{D}}(\widetilde{d}, d) = -\log P(\widetilde{d} = d) \tag{6.2}$$

Where y is the true label, $\widetilde{y}$ is the pseudo-label, d is the domain index and $\widetilde{d}$ is the domain pseudo-label. Therefore, we formulate CAN as a minimax optimization problem with two competitive terms defined as follows:

$$\mathcal{J}_{\mathcal{C}} = \sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{L}_i}[\mathcal{L}_{\mathcal{C}}(\mathcal{C}_i, y)] \tag{6.3}$$

$$\mathcal{J}_{\mathcal{D}} = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}\sim\mathbb{L}_i\cup\mathbb{U}_i}[\mathcal{L}_{\mathcal{D}}(\mathcal{D}([\mathcal{F}_s(\mathbf{x}), \mathcal{C}_i]), d)] \tag{6.4}$$

Where $[\cdot,\cdot]$ is the concatenation of two vectors, $\mathcal{C}_i = \mathcal{C}([\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^i(\mathbf{x})])$ is the prediction probability of the given instance $\mathbf{x}$. $\mathcal{C}$ and $\mathcal{D}$ adopt MLPs with a softmax layer on top. For the domain-specific feature extractors $\{\mathcal{F}_d^i\}_{i=1}^M$, the training is straightforward, as their objective is simple: help $\mathcal{C}$ reduce classification error rates on labeled data. The objective of training $\mathcal{F}_s$ has two-fold: (1) help $\mathcal{C}$ reduce classification error rates on labeled data; (2) confuse $\mathcal{D}$ to reach an optimum.

6.2.1 Entropy Conditioning

We condition the domain discriminator $\mathcal{D}$ on the joint variable $(f, c) = (\mathcal{F}_s(\mathbf{x}), \mathcal{C}_i)$. For brevity, here we use f and c to denote $\mathcal{F}_s(\mathbf{x})$ and $\mathcal{C}_i$, respectively. If different instances have equal importance in adversarial training, the hard-to-transfer instances with uncertain predictions may deteriorate the system performance [98]. To alleviate this harmful effect, we employ the entropy criterion $E(c) = -\sum_{k=1}^{2}[c_k log c_k]$ to quantify the uncertainty of label predictions, where c_k is the probability of predicting an instance to category k (negative: k = 1, positive: k = 2). By using this entropy criterion, we can impose higher priority to easy-to-transfer instances with certain predictions. We reweigh these instances by an entropy-aware term: $w(c) = 1 + e^{-E(c)}$. Therefore, the improved $\mathcal{J}_\mathcal{D}$ is defined as:

$$\mathcal{J}_\mathcal{D}^E = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}\sim\mathbb{L}_i\cup\mathbb{U}_i}[w(c)\mathcal{L}_\mathcal{D}(\mathcal{D}([\mathcal{F}_s(\mathbf{x}), \mathcal{C}_i]), d)] \tag{6.5}$$

In summary, the minimax game of CAN is formulated as:

$$\min_{\mathcal{F}_s, \{\mathcal{F}_d^i\}_{i=1}^M, C} \max_{\mathcal{D}} \quad \mathcal{J}_\mathcal{C} + \lambda \mathcal{J}_\mathcal{D}^E \tag{6.6}$$

Where λ is a hyperparameter balancing the two competing objectives. The entropy conditioning empowers the entropy minimization principle [99] and controls the certainty of the predictions, enabling CAN have the ability to generalize on unseen

domains with no labeled data. The CAN training process is illustrated in Algorithm 3.

Algorithm 3 Stochastic gradient descent training algorithm

1: **Input:** labeled data $\mathbb{L}_i$ and unlabeled data $\mathbb{U}_i$ in M domains; a hyperparameter λ.

2: **for** number of training iterations **do**

3: Sample labeled mini-batches from the multiple domains $B^\ell = \{B_1^\ell, \cdots, B_M^\ell\}$.

4: Sample unlabeled mini-batches from the multiple domains $B^u = \{B_1^u, \cdots, B_M^u\}$.

5: Calculate $loss = \mathcal{J}_\mathcal{C} + \lambda \mathcal{J}_\mathcal{D}^E$ on B^ℓ and B^u;

Update $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^M$, $\mathcal{C}$ by descending along the gradients $\Delta loss$.

6: Calculate $l_D = \mathcal{J}_\mathcal{D}^E$ on B^ℓ and B^u;

Update $\mathcal{D}$ by ascending along the gradients Δl_D.

7: **end for**

6.2.2 Theoretical Analysis

In this section, we present an analysis showing the validity of the CAN approach for MDTC. All proofs are given in the next section. The objective of CAN is equivalent to minimizing the total divergence among the M joint distributions. First, we define different joint distributions as $P_i(f, c) \triangleq P(f = \mathcal{F}_s(\mathbf{x}), c = \mathcal{C}_i | \mathbf{x} \in D_i)$. Combining $\mathcal{L}_\mathcal{D}$ with $\mathcal{J}_\mathcal{D}$, the objective of $\mathcal{D}$ can be written as:

$$\mathcal{J}_\mathcal{D} = -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i([f, c])] \tag{6.7}$$

Where $\mathcal{D}_i([f, c])$ yields the probability of the vector $([f, c])$ coming from the i-th domain. We first derive that CAN could achieve its optimum if and only if all M joint distributions are identical.

Lemma 6.1. *For any given $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^M$ and $\mathcal{C}$, the optimum conditional domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*([f,c]) = \frac{P_i(f,c)}{\sum_{j=1}^M P_j(f,c)} \tag{6.8}$$

Then we provide the main theorem for the CAN framework:

Theorem 6.1. *Let $\widetilde{P}(f,c) = \frac{\sum_{i=1}^M P_i(f,c)}{M}$, when $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, we have:*

$$\mathcal{J}_{\mathcal{D}^*} = M \log M - \sum_{i=1}^{M} KL(P_i(f,c)||\widetilde{P}(f,c)) \tag{6.9}$$

where $KL(\cdot)$ is the Kullback-Leibler (KL) divergence [100] of each joint distribution $P_i(f,c)$ to the centroid $\widetilde{P}(f,c)$.

Finally, considering the non-negativity and convexity of the KL-divergence [101], we have:

Corollary 6.1. *When $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, $\mathcal{J}_{\mathcal{D}^*}$ is $M \log M$. The optimum can be obtained if and only if $P_1(f,c) = P_2(f,c) = ... = P_M(f,c) = \widetilde{P}(f,c)$.*

Therefore, by using conditional adversarial training, we can train the conditional domain discriminator on the joint variable (f,c) to minimize the total divergence across different domains, yielding promising performance on MDTC tasks.

6.3 Proof

Assume there exist M domains, for each domain $\mathcal{D}_i$, we have a joint distribution defined as:

$$P_i(f,c) \triangleq P(f = \mathcal{F}_s(\mathbf{x}), c = \mathcal{C}_i | \mathbf{x} \in \mathcal{D}_i)$$

where $\mathcal{C}_i = \mathcal{C}([\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^i(\mathbf{x})])$ is the prediction probability of the given instance $\mathbf{x}$, $[\cdot, \cdot]$ is the concatenation of two vectors. The objective of $\mathcal{D}$ is to minimize $\mathcal{J}_\mathcal{D}$:

$$\mathcal{J}_\mathcal{D} = -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i([f, c])]$$

where $\mathcal{D}_i([f, c])$ is the probability of the vector $([f, c])$ coming from the i-th domain. Therefore, we have:

$$\sum_{i=1}^{M} \mathcal{D}_i([f, c]) = 1 \tag{6.10}$$

Lemma 6.1. *For any given $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^M$ and $\mathcal{C}$, the optimum conditional domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*([f, c]) = \frac{P_i(f, c)}{\sum_{j=1}^{M} P_j(f, c)}$$

Proof. For any given $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^M$ and $\mathcal{C}$, the optimum

$$\begin{aligned}
\mathcal{D}^* &= \arg\min_{\mathcal{D}} \mathcal{J}_\mathcal{D} \\
&= \arg\min_{\mathcal{D}} -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i([f, c])] \\
&= \arg\max_{\mathcal{D}} \sum_{i=1}^{M} \int_{(f,c)} P_i(f, c) \log \mathcal{D}_i([f, c]) d(f, c) \\
&= \arg\max_{\mathcal{D}} \int_{(f,c)} \sum_{i=1}^{M} P_i(f, c) \log \mathcal{D}_i([f, c]) d(f, c)
\end{aligned}$$

Here, we utilize the Lagrangian Multiplier for $\mathcal{D}^*$ under the condition (6.10). We have:

$$L(\mathcal{D}_1, ..., \mathcal{D}_M, \lambda) = \sum_{i=1}^{M} P_i \log \mathcal{D}_i - \lambda(\sum_{i=1}^{M} \mathcal{D}_i - 1)$$

Let $\nabla L = 0$, we have:

$$\begin{cases} \nabla_{\mathcal{D}_i} \sum_{j=1}^{M} P_j \log \mathcal{D}_j - \lambda \nabla_{\mathcal{D}_i} (\sum_{j=1}^{M} \mathcal{D}_j - 1) = 0 \\ \sum_{i=1}^{M} \mathcal{D}_i = 1 \end{cases}$$

From the two above equations, we have:

$$\mathcal{D}_i^*(f, c) = \frac{P_i(f, c)}{\sum_{j=1}^{M} P_j(f, c)}$$

□

Theorem 6.1. *Let* $\widetilde{P}(f, c) = \frac{\sum_{i=1}^{M} P_i(f,c)}{M}$*, when* $\mathcal{D}$ *is trained to its optimum* $\mathcal{D}^*$*, we have:*

$$\mathcal{J}_{\mathcal{D}^*} = M \log M - \sum_{i=1}^{M} KL(P_i(f, c) || \widetilde{P}(f, c))$$

where $KL(\cdot)$ *is the Kullback-Leibler (KL) divergence of each joint distribution* $P_i(f, c)$ *to the centroid* $\widetilde{P}(f, c)$

Proof. Let $\widetilde{P}(f, c) = \frac{\sum_{i=1}^{M} P_i(f,c)}{M}$. We have:

$$\sum_{i=1}^{M} KL(P_i(f, c) || \widetilde{P}(f, c)) = \sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i} [\log \frac{P_i(f, c)}{\widetilde{P}(f, c)}]$$

When $\mathcal{D}$ is updated to $\mathcal{D}^*$, we have:

$$
\begin{aligned}
\mathcal{J}_{\mathcal{D}^*} &= -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i^*([f,c])] \\
&= -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)}] \\
&= -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)} + \log M] + M \log M \\
&= M \log M - \sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\frac{\sum_{j=1}^{M} P_j(f,c)}{M}}] \\
&= M \log M - \sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\widetilde{P}(f,c)}] \\
&= M \log M - \sum_{i=1}^{M} KL(P_i(f,c)||\widetilde{P}(f,c))
\end{aligned}
$$

□

In our model, a minimax game is implemented to achieve the optimum:

$$\min_{\mathcal{F}_s, \{\mathcal{F}_d^i\}_{i=1}^M, C} \max_{\mathcal{D}} \quad \mathcal{J}_{\mathcal{C}} + \lambda \mathcal{J}_{\mathcal{D}}$$

Therefore, by the non-negativity and convexity of the KL-divergence, we can have the corollary:

Corollary 6.1. *When $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, $\mathcal{J}_{\mathcal{D}^*}$ is $M \log M$. The optimum can be obtained if and only if $P_1(f,c) = P_2(f,c) = ... = P_M(f,c) = \widetilde{P}(f,c)$.*

6.4 Experiments

6.4.1 Implementation Details

All experiments are implemented by using PyTorch. The CAN has one hyperparameter: λ, which is fixed as 1 in all experiments. In the experiments, we set the domain-invariant feature dimension as 128 and domain-specific feature dimension as 64, $\mathcal{C}$ and $\mathcal{D}$ are both MLPs whose input dimensions are 128+64, and 128+2 respectively, they both have one hidden layer whose dimension is the same as the input dimension. We use ReLU as the activation function in each component. The dropout rate for each component is set as 0.4. The Adam optimizer [86] with a learning rate of 0.0001 is used to train the model. We set the batch size as 8 and the number of training iterations as 50. For the experiments on the Amazon review dataset, MLPs with two hidden layers are used as the feature extractors, the two hidden layers contain 1,000 and 500 units respectively. The input dimension of feature extractors is 5,000. For the experiments on the FDU-MTL dataset, CNN with one convolutional layer is used as the feature extractor. Its kernel sizes are (3, 4, 5), and the number of kernels is 200. The input dimension of the convolutional layer is 100, we use the word embedding, obtained by processing each word of the input sequence through word2vec [87], as the input of the feature extractors.

6.4.2 Comparison Methods

We first conduct experiments on multi-domain text classification. The CAN model is compared with a number of state-of-the-art baselines:

- CMSC-LS, CMSC-SVM, CMSC-Log: The collaborative multi-domain sentiment classification method which trains a shareable classifier and a set of domain-specific classifiers for all domains. The final prediction is determined by

the combination of the outputs of the shareable classifier and the corresponding classifier of the given instance. The CMSC model is trained with squared loss, hinge loss, and log loss respectively [85].

- MAN: The multinomial adversarial network for multi-domain text classification [10]. This method uses two forms of loss functions to train the multinomial domain discriminator: The negative log-likelihood loss (MAN-NLL) and the least square loss (MAN-L2). The MAN also induces the convergence conditional for adversarial MDTC with regard to the two loss functions.

- MT-CNN: The deep neural network model learns features across different tasks given very limited prior knowledge by using a single convolutional layer [58].

- MT-DNN: The multi-task deep neural network model with bag-of-words input and MLPs [89], in which a hidden layer is shared.

- ASP-MTL: The adversarial multi-task learning model for text classification [13], in which shared-private paradigm, adversarial training, and an orthogonal restriction are used to guide feature extraction.

- MT-BL: The multi-task learning with bidirectional language [76], which introduces language modeling as an auxiliary task to encourage the domain-specific feature extractors to capture more syntactic and semantic information, and a uniform label distribution-based loss constraint to the shared feature extractor to enhance the ability to learn domain-invariant features.

- DACL: The dual adversarial co-learning method [16], which combines the discriminator-based adversarial training and classifier-based adversarial training to align not only the multiple feature distributions but also the labeled and unlabeled features in the latent space.

Table 7: The classification accuracies of CAN on the Amazon review dataset.

Domain	CMSC-LS	CMSC-SVM	CMSC-Log	MAN-L2	MAN-NLL	DACL	MRAN	CAN
Books	82.10	82.26	81.81	82.46	82.98	83.45	**84.60**	83.76 ± 0.20
DVD	82.40	83.48	83.73	83.98	84.03	85.50	**85.60**	84.68 ± 0.16
Electr.	86.12	86.76	86.67	87.22	87.06	87.40	**89.10**	88.34 ± 0.14
Kit.	87.56	88.20	88.23	88.53	88.57	90.00	**91.25**	90.03 ± 0.19
AVG	84.55	85.18	85.11	85.55	85.66	86.59	**87.64**	86.70 ± 0.11

- MRAN: The mixup regularized adversarial network [102], which incorporates the domain mixup and category mixup regularizer to enrich the intrinsic feature pattern in the shared latent space to boost the transferability and discriminability of the learned features.

All the comparison methods use the standard partitions of the datasets. Thus, we directly cite the results from [10, 76] for fair comparisons.

6.4.3 Results

We run the experiments in 5 trial runs and report the average classification accuracy with standard deviation. The experimental results on the Amazon review dataset and the FDU-MTL dataset are shown in Table 7 and Table 8, respectively. From Table 7, it can be seen that our model yields competitive performance compared to the state-of-the-art baselines.

From the experimental results on the FDU-MTL dataset, reported in Table 8, we observe that the CAN model obtains the best accuracies on 6 of 16 domains and achieves the best result in terms of the average classification accuracy. The experimental results on these two MDTC benchmarks validate the effectiveness of our model.

6.4.4 Multi-Source Unsupervised Domain Adaptation

In the MDTC scenario, the model requires labeled training data from each domain. However, in real-world applications, many domains may have no labeled data at all. Therefore, it is of great importance to evaluate the performance of our proposed CAN approach in extreme cases where the target domain has no labeled data.

In the multi-source unsupervised domain adaptation (MS-UDA) setting, we have multiple source domains with both labeled and unlabeled data and one target domain with only unlabeled data, MS-UDA allows adaptation from multiple source domains to a single target domain. The CAN approach can leverage both labeled and unlabeled data to learn feature representations, it is supposed to be effective in the MS-UDA setting. As the target domain has no labeled data at all, the domain discriminator is updated only on unlabeled data in this setting. When conducting text classification on the target domain, we only feed the shared feature to $\mathcal{C}$ and set the domain-specific feature vector to 0.

We conduct the MS-UDA experiments on the Amazon review dataset. In the experiments, for each of the four separate experiments, three of the four domains are regarded as the source domains, and the remaining one is used as the target one. The evaluations are conducted on the target domain. In order to validate CAN's effectiveness, we compare CAN with several baselines: (1) A MLP trained on the source domains; (2) the marginalized denoising autoencoder (mSDA) [90]; (3) the domain adversarial neural network (DANN) [24]. When training mSDA and DANN, we combine the data from all source domains into a single source domain, ignoring the domain differences. (4) the multi-source domain adaptation neural network (MDAN(H) and MDAN(S)) [91]; (5) the multinomial adversarial network (MAN-L2 and MAN-NLL) [10].

In Table 9, we observe that the CAN model outperforms all the baselines on three

out of four domains. In terms of the average classification accuracy, the CAN method achieves superior performance. This suggests that our model has a good ability to generalize on unseen domains.

6.4.5 Ablation Study

The CAN model adopts the conditional domain discriminator and entropy conditioning. In this section, we investigate how these two components impact the performance of our model on the Amazon review dataset. In particular, three ablation variants are induced: (1) CAN w/o C, the variant of the proposed CAN model without conditioning the domain discriminator on label predictions, which utilizes the standard domain discriminator and entropy conditioning; (2) CAN w/o E, the variant of the proposed CAN model without the entropy conditioning, which hence imposes equal importance to different instances; (3) CAN w/o CE, the variant of the proposed CAN model which only uses standard adversarial training for domain alignment. The results of the ablation study are shown in Table 10, in which we observe that all variants produce inferior results, and the full model produces the best results. In summary, the ablation study suggests that both conditional domain discriminator and entropy conditioning contribute to the CAN model.

6.4.6 Parameter Sensitivity Analysis

The proposed CAN approach has one hyperparameter λ, which is used to balance $\mathcal{J}_{\mathcal{C}}$ and $\mathcal{J}_{\mathcal{D}}^{E}$. We conduct parameter sensitivity analysis on the Amazon review dataset. The λ is evaluated in the range $\{0.0001, 0.001, 0.01, 0.1, 1.0, 10.0\}$. The experimental results are shown in Figure 8. The average classification accuracy across the four domains is reported. It can be noted that from 0.0001 to 1.0, the performance increases with λ increasing, the performance change is very small. Then the accuracy reaches

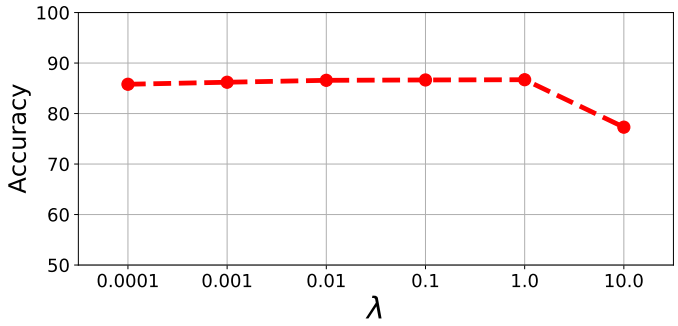


Figure 8: The parameter sensitivity analysis of CAN.

the optimum at the point $\lambda = 1.0$, while the further increase of λ will dramatically deteriorate the performance. This analysis suggests that the selection of λ has an influence on the system performance.

6.5 Summary

In this chapter, we focus on the fourth MDTC challenge shown in Section 1.2: when performing adversarial training on marginal feature distributions, as the class-aware information is ignored, features belonging to different labels can be aligned, resulting in weak discriminability of the learned features. In order to address this issue, we propose conditional adversarial networks (CANs). This approach performs alignment on joint distributions of shared features and label predictions to improve the system performance. The CAN approach adopts the shared-private paradigm, trains the domain discriminator by conditioning it on discriminative information conveyed by the label predictions to encourage the shared feature extractor to capture more discriminative information, and exploits entropy conditioning to guarantee the transferability of the learned features. Experimental results on two MDTC benchmarks demonstrate that the CAN model can not only boost the average classification accuracy for MDTC but also promote the generalization ability when tackling unseen domains. In the next

chapter, we will propose the co-regularized adversarial learning approach for MDTC.

Table 8: The classification accuracies of CAN on the FDU-MTL dataset.

Domain	MT-CNN	MT-DNN	ASP-MTL	MAN-L2	MAN-NLL	MT-BL	DACL	MRAN	CAN
books	84.5	82.2	84.0	87.6	86.8	**89.0**	87.5	87.0	87.8 ± 0.2
electronics	83.2	81.7	86.8	87.4	88.8	90.2	90.3	89.0	$\mathbf{91.6 \pm 0.5}$
dvd	84.0	84.2	85.5	88.1	88.6	88.0	**89.8**	89.0	89.5 ± 0.4
kitchen	83.2	80.7	86.2	89.8	89.9	90.5	91.5	**93.0**	90.8 ± 0.3
apparel	83.7	85.0	87.0	87.6	87.6	87.2	89.5	**91.5**	87.0 ± 0.7
camera	86.0	86.2	89.2	91.4	90.7	89.5	91.5	93.0	$\mathbf{93.5 \pm 0.1}$
health	87.2	85.7	88.2	89.8	89.4	**92.5**	90.5	90.0	90.4 ± 0.6
music	83.7	84.7	82.5	85.9	85.5	86.0	86.3	86.5	$\mathbf{86.9 \pm 0.1}$
toys	89.2	87.7	88.0	90.0	90.4	**92.0**	91.3	86.0	90.0 ± 0.3
video	81.5	85.0	84.5	89.5	**89.6**	88.0	88.5	88.5	88.8 ± 0.4
baby	87.7	88.0	88.2	90.0	90.2	88.7	**92.0**	90.0	$\mathbf{92.0 \pm 0.2}$
magazine	87.7	89.5	92.2	92.5	92.9	92.5	93.8	93.5	$\mathbf{94.5 \pm 0.5}$
software	86.5	85.7	87.2	90.4	90.9	**91.7**	90.5	89.5	90.9 ± 0.2
sports	84.0	83.2	85.7	89.0	89.0	89.5	89.3	90.5	$\mathbf{91.2 \pm 0.7}$
IMDb	86.2	83.2	85.5	86.6	87.0	88.0	87.3	**89.0**	88.5 ± 0.6
MR	74.5	75.5	76.7	76.1	76.7	75.7	76.0	**78.5**	77.1 ± 0.9
AVG	84.5	84.3	86.1	88.2	88.4	88.6	89.1	89.0	$\mathbf{89.4 \pm 0.1}$

Table 9: Multi-source unsupervised domain adaptation results of CAN on the Amazon review dataset.

Domain	MLP	mSDA	DANN	MDAN(H)	MDAN(S)	MAN-L2	MAN-NLL	CAN
Books	76.55	76.98	77.89	78.45	78.63	78.45	77.78	**78.91**
DVD	75.88	78.61	78.86	77.97	80.65	81.57	82.74	**83.37**
Elec.	84.60	81.98	84.91	84.83	**85.34**	83.37	83.75	84.76
Kit.	85.45	84.26	86.39	85.80	86.26	85.57	86.41	**86.75**
AVG	80.46	80.46	82.01	81.76	82.72	82.24	82.67	**83.45**

Table 10: Ablation study analysis of CAN.

Method	Books	DVD	Electr.	Kit.	AVG
CAN (full)	83.76	84.68	88.34	90.03	86.70
CAN w/o C	82.45	84.45	87.30	89.65	85.96
CAN w/o E	83.60	84.80	87.70	89.40	86.38
CAN w/o CE	82.98	84.03	87.06	88.57	85.66

Chapter 7

Co-Regularized Adversarial Learning for Multi-Domain Text Classification

7.1 Introuction

With the advent of deep neural networks (DNNs), natural language processing (NLP) has drawn much attention in past years and obtained remarkable accomplishments [81]. Text classification is a fundamental task in NLP [103] and has attracted constant attention in recent years. It has been widely used to reshape business via understanding customers' emotional tendencies [104] and conduct spam detection [105]. The DNN-based text classification models have obtained impressive achievements in various applications [102, 106]. Unfortunately, these achievements always rely on large amounts of labeled data. In many real-world scenarios, abundant labeled training data is not commonly available and data labeling is always expensive and time-consuming. Moreover, text classification is a highly domain-dependent task [82]. A subtle bias between the training and test data distributions can cause catastrophic performance deterioration. Therefore, it is of great significance to investigate how to improve the classification accuracy of the target domain by leveraging available resources from related domains.

Currently, there exist three streams of techniques to address the above problem: (1) The domain-agnostic method, which combines all available labeled data across multiple domains as the training set, ignoring domain differences. Unfortunately, as described in Section 2.4, text classification is well-known to be domain-dependent. The domain-agnostic methods thus can not produce satisfactory results; (2) Fine-tuning large pre-trained language models [11], such as BERT [6] and OpenAI GPT [12], on the target domain. The large pre-trained models can effectively express context information of a word compared to general word embedding and yield remarkable performance in various NLP tasks [6]; (3) The multi-domain text classification (MDTC), which tackles the scenario where each of multiple domains contains limited amounts of labeled data, but in insufficient to train an efficient classifier. The most recent MDTC methods [7, 10, 11, 16, 19, 102] adopt adversarial training, shared-private paradigm, and entropy minimization to yield state-of-the-art performance. Adversarial training is used to align multiple feature distributions in the shared latent space [24, 91]. Shared-private paradigm divides the latent space into shared latent space and domain-specific latent spaces: the shared-latent space aims to capture domain-invariant features that can be generalized across multiple domains, while the domain-specific feature seeks to learn domain-specific knowledge that can contribute only to their corresponding domain [18]. Entropy minimization is often used to regularize the model output and minimize the uncertainty of predictions on unlabeled data [7, 20]. However, these methods still face three issues: (1) Minimizing domain divergence can not fully guarantee the success of domain alignment; (2) Aligning marginal feature distributions risk in matching features with different labels; (3) Entropy minimization can cause over-confident predictions on unlabeled data, degrading the discriminability of the learned features.

To alleviate the above three issues, in this chapter, we propose a co-regularized adversarial learning (CRAL) approach for multi-domain text classification. The CRAL

method constructs two diverse shared latent spaces and aligns class-conditional feature distributions in each of them to improve the system performance. We then use a regularizer to enforce the two alignments to agree with each other with regard to the predictions on unlabeled data, the regularizer aims to shrink the hypothesis space of possible alignments while preserving the correct set of alignments. Lastly, we incorporate the virtual adversarial training [107] with entropy minimization on unlabeled data to impose consistent predictions in-between training samples, ruling out redundant hypothesis classes that abruptly change predictions in the vicinity of the training data [108]. We carry out the empirical analysis on two MDTC benchmarks: the Amazon review dataset and the FDU-MTL dataset. Our proposed CRAL method beat all state-of-the-art baselines. We also conduct extensive experiments on multi-source unsupervised domain adaptation (MS-UDA) task to demonstrate that our CRAL method has a good ability to generalize on unseen domains. Our contributions are listed as follows:

- We propose a co-regularized adversarial learning (CRAL) method for multi-domain text classification which constructs two diverse shared latent spaces, performs adversarial alignment in each of them, penalizes the disagreements of their predictions on unlabeled data, and incorporates virtual adversarial training with entropy minimization to improve the system performance.

- We demonstrate the effectiveness of the CRAL method on two MDTC benchmarks. The experimental results show that our method outperforms the state-of-the-art methods on both datasets. Further experiments on multi-source unsupervised domain adaptation reveal that our model has a good ability to generalize the learned knowledge to unseen domains.

- We also conduct ablation studies and parameter sensitivity analyses to explore the contributions of different components of the CRAL method and how each

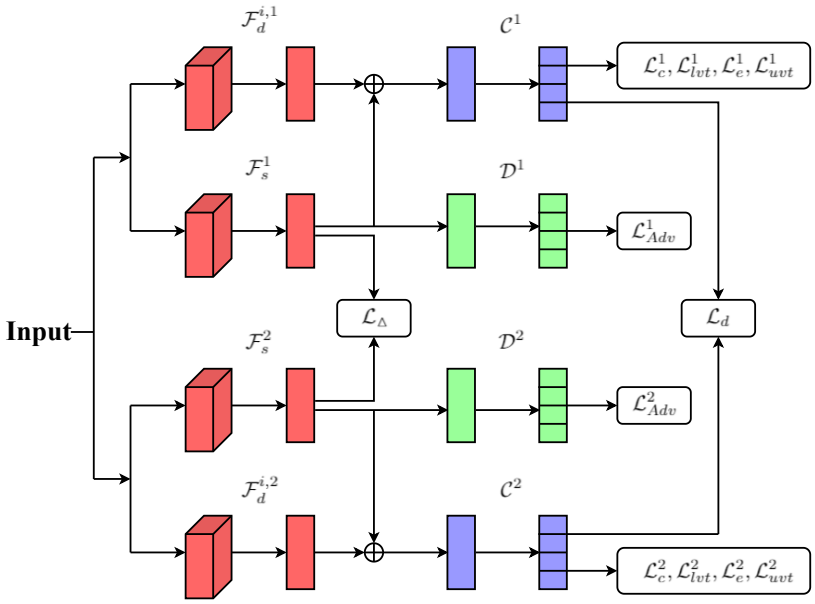


Figure 9: The Architecture of the CRAL framework.

hyperparameter influences the performance of our method.

7.2 Model Architecture

As illustrated in Fig.9, the proposed CRAL model consists of two branches, each branch is composed of four components: a shared feature extractor $\mathcal{F}_s^b$, a set of domain-specific feature extractors $\{\mathcal{F}_d^{i,b}\}_{i=1}^M$, a domain discriminator $\mathcal{D}^b$, and a classifier $\mathcal{C}^b$. b indicates the branch index (e.g $b \in \{1, 2\}$). For simplicity, we ignore the branch index of each component, the following discussion holds for both branches. The shared feature extractor aims to learn domain-invariant features that can generalize across domains. The domain-specific feature extractor captures domain-specific features that are beneficial within their own domain. These feature extractors can adopt the architecture of convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformer, or multi-layer perceptrons (MLPs), depending on the practical task. The feature extractors yield feature representations with a fixed

length. The discriminator aims to distinguish features among different domains, it takes a domain-invariant feature vector as its input. The classifier is used to conduct classification, it uses the concatenation of a domain-invariant feature vector and a domain-specific feature vector as input.

7.3 Multinomial Adversarial Learning

Multinomial adversarial learning is supposed to minimize the domain divergence to implement feature alignment in the shared latent space [7, 16, 102]. In multinomial adversarial learning, the shared feature extractor is trained against the domain discriminator to generate domain-invariant features. In the CRAL method, the multinomial adversarial training is performed independently in each of the two branches. The domain discriminator $\mathcal{D}$ is a M-class classifier, outputting the probabilities of an instance coming from each domain. The classifier $\mathcal{C}$ is a binary classifier, predicting sentiment probabilities. The feature extractors $\mathcal{F}_s$ and $\{\mathcal{F}_d^i\}_{i=1}^M$ are expected to complement each other and maximally capture useful information across domains. The multinomial adversarial learning is encoded as follows:

$$\mathcal{L}_c^b = -\sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x}_i, y_i)\sim\mathbb{L}_i}[y_i^\top \log(\mathcal{G}^{i,b}(\mathbf{x}_i))] \tag{7.1}$$

$$\mathcal{L}_{Adv}^b = -\sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i\sim\mathbb{L}_i\cup\mathbb{U}_i}[d_i^\top \log(\mathcal{D}^b(\mathcal{F}_s^b(\mathbf{x}_i)))] \tag{7.2}$$

Where $\mathcal{L}_c^b$ is the cross-entropy loss for the labeled data, $\mathcal{L}_{Adv}^b$ is the adversarial loss for both labeled and unlabeled data, d_i is the domain index of an instance $\mathbf{x}_i$, and $\mathcal{G}^{i,b}(\mathbf{x}_i) = \mathcal{C}^b([\mathcal{F}_s^b(\mathbf{x}_i), \mathcal{F}_d^{i,b}(\mathbf{x}_i)])$ where $[\cdot,\cdot]$ represents the concatenation of two vectors.

7.4 Co-Regularized Adversarial Learning

Before presenting the co-regularized adversarial learning mechanism, we first present the limitations of existing MDTC methods. The idea of aligning multiple feature distributions for MDTC can be motivated by the theory [10]: we first consider the shared features $\mathbf{f}$ for each domain:

$$P_i(\mathbf{f}) \triangleq P(\mathbf{f} = \mathcal{F}_s(\mathbf{x})|\mathbf{x} \in D_i) \tag{7.3}$$

Where $P(\cdot)$ represents the probability, so we have P_1, P_2, ..., P_M as the M shared feature distributions. Then set $\bar{P} = \frac{\sum_{i=1}^{M} P_i}{M}$ as the centroid of the M feature distributions. The domain discriminator D is expected to be trained to its optimality $\mathcal{D}^*$ if and only if $P_1 = P_2 = ... = P_M = \bar{P}$. Therefore, the objective of MDTC can be regarded as aligning M feature distributions $\{P_i\}_{i=1}^{M}$ to $\bar{P}$ which is equivalent to minimizing the distribution distances between each of the feature distributions and their centroid. Let $\mathcal{H}$ be the hypothesis space, for any $h \in \mathcal{H}$, that maps inputs from the input space X to the label space Y, suppose the mapping function $h = c \circ f$ can be decomposed as the composite of a feature mapping function $f : X \to \mathbb{R}^m$ and a classifier $c : \mathbb{R}^m \to Y$. Then let h' be the labeling function, which can be decomposed as $h' = c' \circ f'$. Motivated by the domain adaptation theory [26], the distance between P_i and $\bar{P}$ can be defined as:

$$d^i_{\mathcal{H}}(P_i, \bar{P}) = 2\text{sup}_{h,h' \in \mathcal{H}} |\mathbb{E}_{\mathbf{f} \sim P_i}[c(\mathbf{f}) \neq c'(\mathbf{f})] - \mathbb{E}_{\mathbf{f} \sim \bar{P}}[c(\mathbf{f}) \neq c'(\mathbf{f})]| \tag{7.4}$$

The MDTC task can be considered as the collection of M domain adaptation tasks between $\{P_i\}_{i=1}^{M}$ and $\bar{P}$, we thus obtain the total distribution distance of MDTC as:

$$d_{\mathcal{H}} = \sum_{i=1}^{M} d^i_{\mathcal{H}}(P_i, \bar{P}) \tag{7.5}$$

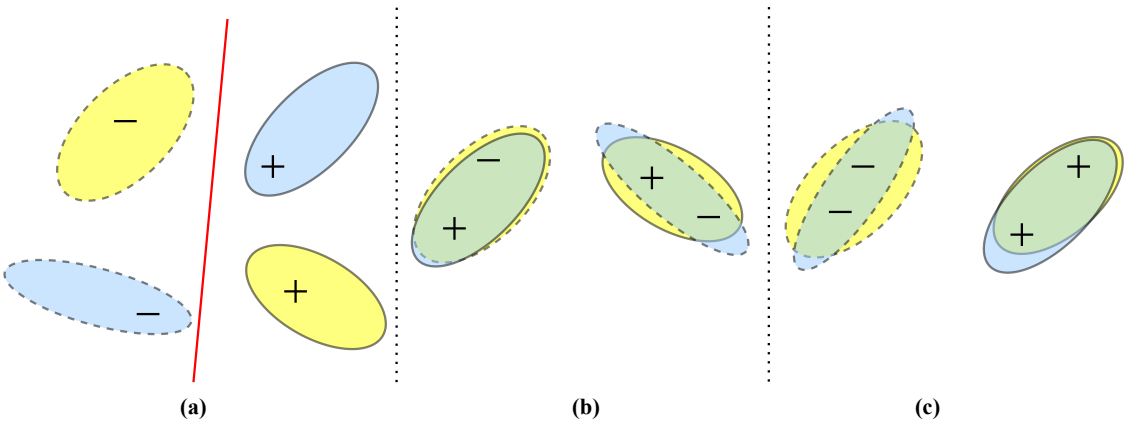


Figure 10: Examples for the domain alignment in MDTC.

As a consequence, the ultimate objective of the MDTC task is to minimize $d_{\mathcal{H}}$. However, a small $d_{\mathcal{H}}$ may not guarantee that the multiple feature distributions are correctly aligned. As illustrated in Fig.10(a), we can observe that $d_{\mathcal{H}}$ can be pushed towards 0 without P_1, P_2, ..., and P_M being aligned correctly. In addition, aligning the marginal feature distributions risks matching features with different labels, resulting in weak discriminability of the learned features (Fig. 10(b)).

To address these issues, the proposed CRAL method conducts class-conditional alignment by learning two classifiers based on two diverse latent spaces ($\mathcal{H}_1$ and $\mathcal{H}_2$) and penalizing the disagreement between their outputs. For any $h_1 \in \mathcal{H}_1$, $h_2 \in \mathcal{H}_2$, to quantify the disagreement between the two classifiers, we exploit the discrepancy between the predictions $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ on unlabeled data among the M domains. When the disagreement approaches zero, it suggests that the domain alignments performed in these two latent spaces have similar feature structures and the M feature distributions are correctly aligned. We here use the ℓ_1 norm to encode the disagreement:

$$\mathcal{L}_d = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i \sim \mathbb{U}_i} ||\mathcal{G}^{i,1}(\mathbf{x}_i) - \mathcal{G}^{i,2}(\mathbf{x}_i)||_1 \tag{7.6}$$

To ensure the sufficient diversity of the two shared latent spaces, we adopt a regularizer formulated as:

$$\mathcal{L}_{\triangle} = \min(\gamma, ||\frac{1}{M} \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i \sim \mathbb{L}_i} (\mathcal{F}_s^1(\mathbf{x}_i) - \mathcal{F}_s^2(\mathbf{x}_i))||_2^2) \tag{7.7}$$

Where $||\cdot||_2^2$ is the ℓ_2 norm, the hyperparameter γ is positive and controls the maximal disparity between these two shared latent spaces. Maximizing $\mathcal{L}_{\triangle}$ is equivalent to pushing the centroids of these two shared latent spaces far apart.

7.5 Virtual Adversarial Training

In the CRAL approach, we also incorporate virtual adversarial training (VAT) with entropy minimization on unlabeled data to impose consistent prediction constraints. In MDTC, entropy minimization is often utilized to control the uncertainty of the predictions. The entropy minimization is formulated as follows:

$$\mathcal{L}_e^b = -\sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i \sim \mathbb{U}_i} [\mathcal{G}^{i,b}(\mathbf{x}_i)^\top \log(\mathcal{G}^{i,b}(\mathbf{x}_i))] \tag{7.8}$$

Even though the entropy minimization could excavate additional feature patterns from unlabeled data to boost the discriminability of the learned features, it also needs to be noted that entropy minimization may bring in overfitting to the unlabeled data if the model has infinite capacity [109]. The over-confident predictions on unlabeled data can drive the decision boundary across the high-density regions in the latent space [110], degrading the discriminability of the learned features. In particular, as

unlabeled data have no supervision, the model may abruptly change its predictions in the vicinity of the unlabeled data [111] (e.g. for any $h \in \mathcal{H}$, $h(u+\epsilon) \neq h(u)$, where small perturbations ϵ occur to unlabeled data points u). To avoid this risk, virtual adversarial training (VAT) with entropy minimization is incorporated into our CRAL model. The VAT technique can smooth the prediction surface around the unlabeled data points. VAT aims to search for small perturbations ϵ that maximize the change of the predictions on unlabeled data, enforcing consistent predictions to the model [107]. Following [109], we also apply VAT to labeled data. The VAT losses on unlabeled data $\mathcal{L}^b_{uvt}$ and labeled data $\mathcal{L}^b_{lvt}$ are formulated as:

$$\mathcal{L}^b_{uvt} = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i \sim \mathbb{U}_i}[\max_{||r|| \leq \epsilon} D_{kl}(\mathcal{G}^{i,b}(\mathbf{x}_i)||\mathcal{G}^{i,b}(\mathbf{x}_i + r))] \tag{7.9}$$

$$\mathcal{L}^b_{lvt} = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}_i \sim \mathbb{L}_i}[\max_{||r|| \leq \epsilon} D_{kl}(\mathcal{G}^{i,b}(\mathbf{x}_i)||\mathcal{G}^{i,b}(\mathbf{x}_i + r))] \tag{7.10}$$

Where $D_{kl}(\cdot||\cdot)$ is the Kullback–Leibler (KL) divergence [112]. To calculate the KL divergence in practice, we always need integrals, which is almost impossible. Thus, we use chi-square to approximate the KL-divergence in implementation.

7.6 The Final Objective

The final objective function of the CRAL method is formulated as:

$$\begin{aligned} \min_{\mathcal{F}^b_s, \{\mathcal{F}^{i,b}_d\}, \mathcal{C}^b} \max_{\mathcal{D}^b} \sum_{b=1}^{2} [\mathcal{L}^b_c + \lambda_{Adv} * \mathcal{L}^b_{Adv} + \lambda_{lvt} * \mathcal{L}^b_{lvt} \\ + \lambda_{uvt} * (\mathcal{L}^b_e + \mathcal{L}^b_{uvt})] + \lambda_d * \mathcal{L}_d - \lambda_\triangle * \mathcal{L}_\triangle \end{aligned} \tag{7.11}$$

The CRAL model is trained with mini-batch stochastic gradient descent and in an alternating fashion following [21], the training algorithm of CRAL is presented in

Algorithm 1. In each iteration, the input data should be fed into the two branches to generate the predictions. The CRAL mechanism boosts the system performance by punishing the disagreement of predictions induced from the two diverse adversarial training streams and utilizing the VAT with entropy minimization to enforce the consistency regularization to the model. γ is a hyperparameter that controls the diversity extent of two shared latent spaces. λ_{Adv}, λ_d, $\lambda_\triangle$, λ_{uvt} and λ_{lvt} are hyperparameters that balance different loss functions. In our experiments, we set γ, λ_{Adv}, λ_d and $\lambda_\triangle$ as 10, 1, 0.00001 and 0.0001. According to our parameter sensitivity analysis, the selections of λ_{uvt} and λ_{lvt} can significantly influence the performance of our model. Thus, these two hyperparameters need to be tuned for each benchmark.

Algorithm 4 SGD training algorithm

1: **Input:** labeled data $\mathbb{L}_i$ and unlabeled data $\mathbb{U}_i$ in M domains; hyperparameters: γ, λ_{Adv}, λ_d, $\lambda_\triangle$, λ_{uvt}, λ_{lvt}

2: **for** number of training iterations **do**

3: Sample labeled mini-batches from the multiple domains $B^\ell = \{B_1^\ell, \cdots, B_M^\ell\}$.

4: Sample unlabeled mini-batches from the multiple domains $B^u = \{B_1^u, \cdots, B_M^u\}$.

5: Calculate $l_D = \sum_{b=1}^{2} \lambda_{Adv} \mathcal{L}_{Adv}^b$ on B^ℓ and B^u;
Update $\mathcal{D}^1$ and $\mathcal{D}^2$ by ascending along gradients ∇l_D.

6: Calculate $loss = \sum_{b=1}^{2} [\mathcal{L}_c^b + \lambda_{Adv} * \mathcal{L}_{Adv}^b + \lambda_{uvt} * (\mathcal{L}_e^b + \mathcal{L}_{uvt}^b) + \lambda_{lvt} * \mathcal{L}_{lvt}^b] + \lambda_d * \mathcal{L}_d - \lambda_\triangle * \mathcal{L}_\triangle$ on B^ℓ and B^u;
Update $\mathcal{F}_s^1$, $\mathcal{F}_s^2$, $\{\mathcal{F}_d^{i,1}\}$, $\{\mathcal{F}_d^{i,2}\}$, $\mathcal{C}^1$, $\mathcal{C}^2$ by descending along gradients $\nabla loss$.

7: **end for**

7.7 Discussion

In MDTC, there exist three ultimate objectives to accomplish: (1) Minimizing the error rate on labeled data; (2) Minimizing the uncertainty of predictions on unlabeled data; (3) Aligning the multiple feature distributions in the shared latent space such that the original features can be transformed into domain-invariant. In general, the first and second objectives aim to impose discriminability on the learned features, while the third one imposes transferability to the learned features. The first objective can be simply achieved by minimizing the classification loss on labeled data. In past decades, MDTC methods often focus on improving the procedure of adversarial training to optimize the adversarial alignment [16, 102], few works explore how to optimize the uncertainty of predictions on unlabeled data to make advance. Semi-supervised learning suggests that unlabeled data can be used to explore additional feature patterns in the shared latent space [99]. For instance, cluster structures in the distributions could hint at the separation of samples into different labels. This is termed the cluster assumption: if two samples reside in the same cluster, they are likely to belong to the same class. The cluster assumption encourages the decision boundary to lie in the low-density regions of the distributions. The intuition is simple: If a decision boundary lies in a high-density region, it will cut the cluster into two different classes, enabling samples lying in the same cluster to have two different labels. The above phenomenon is named the violation of the cluster assumption. By penalizing the violation of the cluster assumption, we can push the decision boundary away from the high-density regions of the distributions, imposing consistent prediction constraints on the model.

In our approach, we improve the adversarial training by constructing two adversarial training branches, maximizing the diversity of their shared latent spaces while

minimizing the disagreement of their predictions on unlabeled data. This improvement can shrink the search space of possible alignments while preserving the correct set of alignments. Moreover, we optimize the uncertainty of predictions on unlabeled data by employing the VAT with entropy minimization. The VAT can impose consistency regularization on the training data by penalizing the violation of the cluster assumption.

7.8 Experiments

7.8.1 Implementation Details

We implemented our approach using Pytorch. The CRAL has six parameters: γ, λ_{Adv}, λ_d, $\lambda_\triangle$, λ_{uvt} and λ_{lvt}. In the experiments, we fix $\gamma = 10$, $\lambda_{Adv} = 1$, $\lambda_d = 0.00001$, $\lambda_\triangle = 0.0001$, λ_{uvt} and λ_{lvt} are selected in the range $\{0.001, 0.01, 0.1, 1, 3, 10\}$. The proposed MRAN has four hyperparameters, λ_d, λ_a, λ_u and λ_m. In the experiments, we fixed $\lambda_d = 1$, $\lambda_a = 0.001$, $\lambda_u = 0.1$ and $\lambda_m = 0.00001$. In the experiments, we set the domain-invariant feature dimension as 128 and domain-specific feature dimension as 64, the domain discriminators and classifiers are all MLPs whose input dimensions are 128+64 and 128 respectively, they all have one hidden layer whose dimension is the same as the input dimension. We use ReLU as the activation function in each component. The dropout rate for each component is set as 0.4. The Adam optimizer [86] with a learning rate of 0.0001 is used to train the model. We set the batch size as 8 and the number of training iterations as 50. For the experiments on the Amazon review dataset, MLPs with two hidden layers are used as the feature extractors, the two hidden layers contain 1,000 and 500 units respectively. The input dimension of feature extractors is 5,000. For the experiments on the FDU-MTL dataset, CNN with one convolutional layer is used as the feature extractor. Its kernel sizes are (3,

4, 5), and the number of kernels is 200. The input dimension of the convolutional layer is 100, we use the word embedding, obtained by processing each word of the input sequence through word2vec [87], as the input of the feature extractors. For the evaluation, we use the average prediction probability scores of the two classifiers to make the final prediction of each instance.

7.8.2 Comparison Methods

We compare our MRAN model with several state-of-the-art baselines:

- CMSC-LS, CMSC-SVM, CMSC-Log: The collaborative multi-domain sentiment classification method which trains a shareable classifier and a set of domain-specific classifiers for all domains. The final prediction is determined by the combination of the outputs of the shareable classifier and the corresponding classifier of the given instance. The CMSC model is trained with squared loss, hinge loss, and log loss respectively [85].

- MAN: The multinomial adversarial network for multi-domain text classification [10]. This method uses two forms of loss functions to train the multinomial domain discriminator: The negative log-likelihood loss (MAN-NLL) and the least square loss (MAN-L2). The MAN also induces the convergence conditional for adversarial MDTC with regard to the two loss functions.

- MT-CNN: The deep neural network model learns features across different tasks given very limited prior knowledge by using a single convolutional layer [58].

- MT-DNN: The multi-task deep neural network model with bag-of-words input and MLPs [89], in which a hidden layer is shared.

- ASP-MTL: The adversarial multi-task learning model for text classification [13],

in which shared-private paradigm, adversarial training, and an orthogonal restriction are used to guide feature extraction.

- MT-BL: The multi-task learning with bidirectional language [76], which introduces language modeling as an auxiliary task to encourage the domain-specific feature extractors to capture more syntactic and semantic information, and a uniform label distribution-based loss constraint to the shared feature extractor to enhance the ability to learn domain-invariant features.

- DACL: The dual adversarial co-learning method [16], which combines the discriminator-based adversarial training and classifier-based adversarial training to align not only the multiple feature distributions but also the labeled and unlabeled features in the latent space.

- DA-MTL: The dynamic attentive sentence encoding method [11], which utilizes the attention mechanism and introduces a task-dependent query vector to learn domain-specific features to enhance the discriminability of the domain-invariant features.

- CLR-MTL: The global and local shared representation-based generic dual-channel multi-task learning method [77], which deploys adversarial training and mixture-of-experts on two separate channels to capture global-shared, local-shared, and private features simultaneously.

- MRAN: The mixup regularized adversarial network [102], which incorporates the domain mixup and category mixup regularizer to enrich the intrinsic feature pattern in the shared latent space to boost the transferability and discriminability of the learned features.

Table 11: The classification accuracies of CRAL on the Amazon review dataset.

Domain	CMSC-LS	CMSC-SVM	CMSC-Log	MAN-L2	MAN-NLL	DACL	MRAN	CAN	CRAL
Books	82.10	82.26	81.81	82.46	82.98	83.45	84.60	83.76	$\mathbf{85.26 \pm 0.13}$
DVD	82.40	83.48	83.73	83.98	84.03	85.50	85.60	84.68	$\mathbf{85.83 \pm 0.14}$
Electr.	86.12	86.76	86.67	87.22	87.06	87.40	89.10	88.34	$\mathbf{89.32 \pm 0.09}$
Kit.	87.56	88.20	88.23	88.53	88.57	90.00	91.25	90.03	$\mathbf{91.60 \pm 0.17}$
AVG	84.55	85.18	85.11	85.55	85.66	86.59	87.64	86.70	$\mathbf{88.00 \pm 0.12}$

- CAN: The conditional adversarial network [7], which aligns the joint distributions of domain-invariant features and label predictions to improve the discriminability of the learned features. The entropy conditioning mechanism is also incorporated to guarantee the transferability of the learned features.

All the comparison methods use the standard partitions of the datasets. Thus, we cite the results from [10, 19, 76] for fair comparisons.

7.8.3 Results

Multi-Domain Text Classification

We run the experiments in 5 trial runs and report the average classification accuracy with standard deviation. The experimental results on the Amazon review dataset and FDU-MTL dataset are reported in Table 11 and Table 12, respectively. From Table 11, it can be noted that the CRAL method can not only outperform other baselines on all four domains, but also obtain the best performance in terms of the average classification accuracy. Moreover, it beats the second-best approach MRAN by the margin of 0.36% in terms of the average accuracy.

As shown in Table 12, when we conduct experiments on the more challenging FDU-MTL dataset, we can observe that the CRAL method obtains the best classification

Table 12: The classification accuracies of CRAL on the FDU-MTL dataset.

Domain	MT-CNN	MT-DNN	ASP-MTL	BERT	MAN-L2	MAN-NLL	DA-MTL	DACL	GLR-MTL	MRAN	CAN	CRAL
books	84.5	82.2	84.0	87.0	87.6	86.8	88.5	87.5	88.3	87.0	87.8	**89.3 ± 0.3**
electronics	83.2	88.3	86.8	88.3	87.4	88.8	89.0	90.3	90.3	89.0	**91.6**	89.1±0.5
dvd	84.0	84.2	85.5	85.6	88.1	88.6	88.0	89.8	87.3	89.0	89.5	**91.0 ± 0.2**
kitchen	83.2	80.7	86.2	91.0	89.8	89.9	89.0	91.5	89.8	**93.0**	90.8	92.3 ± 0.2
apparel	83.7	85.0	87.0	90.0	87.6	87.6	88.8	89.5	88.2	91.5	87.0	**91.6 ± 0.4**
camera	86.0	86.2	89.2	90.0	91.4	90.7	91.8	91.5	89.5	93.0	93.5	**96.3 ± 0.2**
health	87.2	85.7	88.2	88.3	89.8	89.4	90.3	**90.5**	**90.5**	90.0	90.4	87.8±0.4
music	83.7	84.7	82.5	86.8	85.9	85.5	85.0	86.3	87.5	86.5	86.9	**88.1 ± 0.1**
toys	89.2	87.7	88.0	91.3	90.0	90.4	89.5	91.3	89.8	86.0	90.0	**91.6 ± 0.3**
video	81.5	85.0	84.5	88.0	89.5	89.6	89.5	88.5	90.8	88.5	88.8	**92.6 ± 0.4**
baby	87.7	88.0	88.2	91.5	90.0	90.2	90.5	92.0	**92.3**	90.0	92.0	90.9±0.2
magazine	87.7	89.5	92.2	92.8	92.5	92.9	92.0	93.8	92.3	93.5	94.5	**95.2 ± 0.4**
software	86.5	85.7	87.2	89.3	90.4	90.9	90.8	90.5	**91.8**	89.5	90.9	87.7±0.4
sports	84.0	83.2	85.7	90.8	89.0	89.0	89.8	89.3	87.8	90.5	91.2	**91.3 ± 0.3**
IMDb	86.2	83.2	85.5	85.8	86.6	87.0	89.8	87.3	87.5	89.0	88.5	**90.8 ± 0.3**
MR	74.5	75.5	76.7	74.0	76.1	76.7	75.5	76.0	72.7	**78.5**	77.1	77.3 ± 0.5
AVG	84.5	84.3	86.1	88.1	88.2	88.4	88.2	89.1	88.5	89.0	89.4	**90.2 ± 0.2**

accuracy on 10 out of 16 domains, and yield the best average classification accuracy, outperforming the second-best baseline CAN by 0.8%. The results shown in Table 11 and Table 12 both demonstrate the effectiveness of our proposed CRAL method in MDTC.

Multi-Source Unsupervised Domain Adaptation

When tackling MDTC tasks, the model requires labeled data for each domain. While in real-world applications, many domains may not have any labeled data. Therefore, it is necessary to explore the generalization ability of our method when applied to an unseen domain without labeled data. In the multi-source unsupervised domain adaptation (MS-UDA) experimental setting, we have multiple source domains with

both labeled and unlabeled data and one target domain with unlabeled data, MS-UDA allows adaptation from multiple source domains to a single target domain. Since the MRAN can leverage unlabeled data across different domains to do representation learning, it is supposed to be effective in the MS-UDA setting. Since the target domain has no labeled data, we only feed the shared features to $\mathcal{C}$, while the domain-specific part is set to 0.

The baselines used in the MS-UDA task include: (1) A MLP trained on the source domains. (2) Two single-source domain adaptation methods, marginalized denoising autoencoder (mSDA) [90] and domain adversarial neural network (DANN) [24]. When training mSDA and DANN, we combine the data from all source domains into a single source domain. (3) The state-of-the-art MS-UDA methods, multi-source domain adaptation neural networks (MDAN(H) and MADN(S)) [91] and MAN (MAN-L2 and MAN-NLL) [10].

We conduct the MS-UDA experiments on the Amazon review dataset, following the same setting as [10]. For each of four separate experiments, three out of the four domains are used as the source domains, and the remaining one is used as the target domain. From Table 13, we can see that the proposed CRAL method can outperform other baselines not only on each individual domain, but also in terms of the average accuracy. For the average classification accuracy, our CRAL method outperforms the MAN-NLL method by a margin of 3.00%, suggesting that our CRAL method has a good capacity for transferring knowledge to unseen domains.

7.8.4 Further Analysis

Ablation Study

In order to verify how each component of the CRAL method can impact the performance, we conduct an ablation study on the Amazon review dataset. In particular,

Table 13: The Multi-source unsupervised domain adaptation results of CRAL on the Amazon review dataset.

Domain	mSDA	DANN	MDAN(H)	MDAN(S)	MAN-L2	MAN-NLL	CRAL
Books	76.98	77.89	78.45	78.63	78.45	77.78	**82.49**
DVD	78.61	78.86	77.97	80.65	81.57	82.74	**84.30**
Elec.	81.98	84.91	84.83	85.34	83.37	83.75	**86.82**
Kit.	84.26	86.39	85.80	86.26	85.57	86.41	**89.08**
AVG	80.46	82.01	81.76	82.72	82.24	82.67	**85.67**

we investigate four variants: (1) CRAL w/o $\mathcal{L}_d$, the variant of CRAL without penalizing the disagreement of predictions on the unlabeled data; (2) CRAL w/o $\mathcal{L}_{\triangle}$, the variant of CRAL without enforcing diversity on two shared feature extractors; (3) CRAL w/o $\mathcal{L}_{uvt}$, the variant of CRAL without the VAT on the unlabeled data; (4) CRAL w/o $\mathcal{L}_{lvt}$, the variant of CRAL without the VAT on the labeled data. The comparison results are presented in Table 14. We can see that all four variants induce inferior performance, and the full model produces the best results, validating that all these components contribute to the performance improvement of our model. Moreover, Compared with $\mathcal{L}_{\triangle}$, $\mathcal{L}_{uvt}$ and $\mathcal{L}_{lvt}$, $\mathcal{L}_d$ makes the most significant contribution to the system performance by a margin of 2.13%, illustrating the effectiveness of our co-regularized adversarial learning mechanism.

Parameter Sensitivity Analysis

In this section, we explore the sensitivity of our approach to the values of hyperparameters λ_d, $\lambda_{\triangle}$, λ_{uvt} and λ_{lvt}. These hyperparameters are used to trade-off different loss functions. We conduct the parameter sensitivity analysis on the Amazon review

Table 14: Ablation study of CRAL.

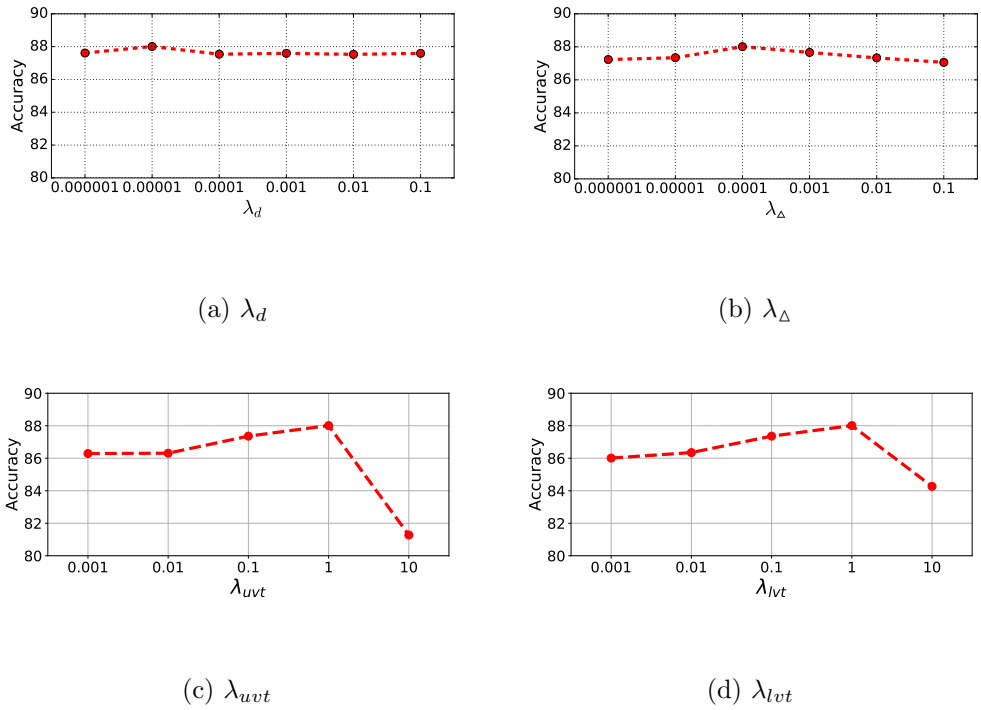
Method	Books	DVD	Electr.	Kit.	AVG
CRAL (full)	85.26	85.83	89.32	91.60	88.00
CRAL w/o $\mathcal{L}_d$	83.08	84.19	87.45	88.74	85.87
CRAL w/o $\mathcal{L}_\triangle$	84.62	85.18	89.15	91.17	87.53
CRAL w/o $\mathcal{L}_{uvt}$	84.25	85.07	88.64	90.88	87.21
CRAL w/o $\mathcal{L}_{lvt}$	84.96	85.35	88.93	91.21	87.61

dataset. When evaluating one hyperparameter, the others are fixed to their default values (e.g. λ_d=0.00001, $\lambda_\triangle$=0.0001, λ_{uvt}=1, λ_{lvt}=1). λ_d and $\lambda_\triangle$ are tested in the range {0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1}, and λ_{uvt} and λ_{lvt} are evaluated in the range {0.001, 0.01, 0.1, 1, 10}. The experimental results are shown in Fig. 11. We report the average classification accuracy.

From Fig. 11, it can be noted that different values of λ_d and $\lambda_\triangle$ cannot significantly influence the system performance. Consequently, we can set λ_d and $\lambda_\triangle$ as 0.00001 and 0.0001 in all experiments. The trends of performance change for λ_{uvt} and λ_{lvt} are similar: with the increase of λ_{uvt} and λ_{lvt}, the classification accuracy first increases and reaches the optimum, then it decreases rapidly. This analysis illustrates that proper selections of λ_{uvt} and λ_{lvt} can effectively improve the performance of our model.

7.9 Summary

In this chapter, we focus on the fourth and fifth MDTC challenges shown in Section 1.2: when performing adversarial training on marginal feature distributions, as the class-aware information is ignored, features belonging to different labels can be



(a) λ_d (b) λ_Δ

(c) λ_{uvt} (d) λ_{lvt}

Figure 11: Parameter sensitivity analysis of CRAL

aligned, resulting in weak discriminability of the learned features; Entropy minimization may encourage the model to make over-confident predictions on unlabeled data, deteriorating the discriminability of the learned features. To address these issues, we propose a co-regularized adversarial learning (CRAL) mechanism for MDTC. This approach constructs two diverse shared latent spaces, performs domain alignment in each of them, and punishes the disagreements of these two alignments with respect to the predictions on unlabeled data. Moreover, virtual adversarial training (VAT) with entropy minimization is incorporated to impose consistency regularization to the CRAL method. Experiments show that our model outperforms state-of-the-art methods on two MDTC benchmarks. In the next chapter, we will propose the margin discrepancy-based adversarial learning method for MDTC.

Chapter 8

Margin Discrepancy-based Adversarial Training for Multi-Domain Text Classification

8.1 Introduction

As mentioned in Section 1.1, multi-domain text classification (MDTC) is proposed to not only train models on labeled text data but also consider relatedness across multiple domains [16]. MDTC seeks to simultaneously utilize all available data across multiple domains to improve the classification accuracy of each domain. Recently, DNN-based models have been widely used for multi-domain learning in both computer vision [83, 84] and NLP [10, 85]. Most recent state-of-the-art MDTC methods adopt the shared-private paradigm [18] and adversarial training [21] to improve the system performance. The adversarial training is used to align multiple feature distributions in the shared latent space to generate domain-invariant features [10]. The shared-private paradigm captures both domain-invariant features and domain-specific features, and enables these two forms of features to complement each other [19]. The idea behind integrating adversarial training and shared-private paradigm with MDTC is to

perform domain alignment across domains and guarantee the optimum separations among the shared latent space and multiple domain-specific feature spaces, preventing domain-specific features from emerging into the shared latent space. Although the most recent MDTC methods design sophisticated algorithms to advance system performance, there is a non-negligible gap between MDTC algorithms and theories: there only exist a few works that explored MDTC from the theoretical perspective. To date, only multinomial adversarial network (MAN) theoretically derives the convergence conditions for adversarial MDTC methods with regard to the least square loss and the negative log-likelihood loss [10]. No work provides detailed theoretical guarantees for MDTC algorithms. Therefore, it is of great significance to derive a generalization bound for the MDTC method such that we can advance the MDTC techniques from a theoretical perspective.

To bridge the gap between algorithms and theories, in this chapter, we first provide the theoretical analysis for incorporating the margin discrepancy in gauging domain divergence in MDTC. The margin discrepancy is a well-defined measurement to gauge divergence between two domains in domain adaptation [113], we extend its usage to the multi-domain scenario. In addition, we derive a new generalization bound based on the margin discrepancy for MDTC by extending the domain adaptation theories [26]. Then according to the provided theoretical analysis, we propose a margin discrepancy-based adversarial training (MDAT) method for multi-domain text classification. The empirical analyses are carried out on two MDTC benchmarks to evaluate the efficacy of the proposed MDAT method. The contributions of this chapter are listed as follows:

- We propose a margin discrepancy-based adversarial training (MDAT) method which employs the margin discrepancy to encode the domain divergence and guide the adversarial training.

- We provide a theoretical justification to support the proposed MDAT method. As far as we know, MDAT is the first attempt to induce the generalization bound for MDTC.

- We evaluate the effectiveness of the MDAT method on two MDTC benchmarks. We also carry out the multi-source unsupervised domain adaptation method to demonstrate that our MDAT method has the ability to generalize to unseen domains.

8.2 Preliminaries

This section first presents some basic notations of MDTC. Second, we give the convergence conditions for adversarial MDTC. Third, we review the measurement of domain divergence for domain adaptation. Lastly, we show details of the margin discrepancy.

8.2.1 Basic Notation

In supervised machine learning setting, the model is trained on a dataset $\widehat{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an input space and $\mathcal{Y}$ is an output space. In particular, $\mathcal{Y}$ is $\{1, ..., k\}$ where k is the number of classes. The samples in $\widehat{D}$ are independently and identically distributed (i.i.d.) samples drawn from the corresponding distribution D.

In the MDTC setting, assume there exist M different datasets $\{\widehat{D}_i\}_{i=1}^{M}$, each $\widehat{D}_i$ consists of two parts: a limited amount of labeled samples $\mathbb{L}_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^{l_i}$, and a large amount of unlabeled samples $\mathbb{U}_i = \{\mathbf{x}_j\}_{j=1}^{u_i}$, where l_i is the number of labeled samples and u_i is the number of unlabeled samples. Each dataset $\widehat{D}_i$ contains n_i samples where $n_i = l_i + u_i$. The samples of dataset $\widehat{D}_i$ are i.i.d. samples drawn from the distribution D_i. The objective of MDTC is to leverage all available resources

from existing domains to improve the system performance, which is measured as the average classification accuracy across the M domains.

Following the notations of [114], we consider the classification task with the hypothesis space $\mathcal{F}$ of scoring functions $f : \mathcal{X} \mapsto \mathbb{R}^{|\mathcal{Y}|} = \mathbb{R}^k$, where the outputs on each dimension indicate the confidence of the prediction. In MDTC, we mainly deal with the binary labels (positive and negative), i.e. $\mathbb{R}^{|\mathcal{Y}|} = \mathbb{R}^2$. The prediction result with an instance $\mathbf{x}$ is the one leads to the largest score $f(\mathbf{x}, y)$. Thus a labeling function space $\mathcal{H}$ containing h_f from $\mathcal{X}$ to $\mathcal{Y}$ can be induced:

$$h_f : \mathbf{x} \mapsto \arg\max_{y \in \mathcal{Y}} f(\mathbf{x}, y) \tag{8.1}$$

The expected error rate and the empirical error rate of a hypothesis class $h \in \mathcal{H}$ with regard to the distribution D and its corresponding dataset $\widehat{D}$ are given by:

$$err_D(h) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim D} \mathbb{1}[(h(\mathbf{x}) \neq y)] \tag{8.2}$$

$$err_{\widehat{D}}(h) \triangleq \mathbb{E}_{(\mathbf{x}_i,y_i)\sim \widehat{D}} \mathbb{1}[(h(\mathbf{x}_i) \neq y_i)] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[h(\mathbf{x}_i) \neq y_i] \tag{8.3}$$

where $\mathbb{1}[a]$ is the indicator function, which is 1 if a is true and 0 otherwise. For brevity, we only show the expected equations in the following.

8.2.2 Adversarial Multi-Domain Text Classification

Adversarial training is first proposed for image generation, then it is extended to align different distributions for domain adaptation and multi-domain text classification [7, 24]. In adversarial training, the f-divergence is often used as the measurement for domain divergence [10, 26]. The multinomial adversarial network (MAN) [10] theoretically derives the convergence condition for MDTC with regard to the negative

log-likelihood loss and the least square loss: The distribution of the shared features $\mathbf{g}$ extracted from the distribution D_i is defined as:

$$P_i(\mathbf{g}) \triangleq P(\mathbf{g} = \mathcal{F}_s(\mathbf{x})|\mathbf{x} \in D_i) \tag{8.4}$$

Where $\mathcal{F}_s(\cdot)$ represents the shared feature embedding function and $P(\cdot)$ represents the probability. Then $P_1, P_2, ..., P_M$ are the M shared feature distributions. Let $\bar{P} = \frac{\sum_{i=1}^{M} P_i}{M}$ be the centroid of the M feature distributions. [10] theoretically proves that the condition where an adversarial MDTC method reaches its optimum is $P_1 = P_2 = ... = P_M = \bar{P}$. Thus, we can tackle the MDTC task by minimizing the total f-divergence between each of the shared feature distributions of the M domains and the centroid of the M feature distributions.

8.2.3 Divergence Measurements

In this section, we review the measurement of distribution divergence in domain adaptation. In the seminal work of [34], given two hypothesis classes $h, h' \in \mathcal{H}$, the $\mathcal{H}\Delta\mathcal{H}$ divergence between two distributions P and Q is defined as:

$$d_{\mathcal{H}\Delta\mathcal{H}} = \sup_{h,h' \in \mathcal{H}} |\mathbb{E}_Q \mathbb{1}[h' \neq h] - \mathbb{E}_P \mathbb{1}[h' \neq h]| \tag{8.5}$$

[83] extends the $\mathcal{H}\Delta\mathcal{H}$ divergence to general loss functions, leading to the discrepancy divergence:

$$dis_{\mathcal{L}} = \sup_{h,h' \in \mathcal{H}} |\mathbb{E}_Q \mathcal{L}(h', h) - \mathbb{E}_P \mathcal{L}(h', h)| \tag{8.6}$$

where $\mathcal{L}$ is a loss function. With the above divergence measures, [34, 83, 113] induce generalization bounds for domain adaptation based on VC-dimension and

Rademacher complexity. These theoretical breakthroughs made significant advancements in algorithm design for domain adaptation.

8.2.4 Margin Discrepancy

The margin discrepancy is proposed to gauge the domain divergence for domain adaptation by using the margin loss [113]. The margin loss is used to achieve strong generalization performance between samples and the classification surface in supervised classification [115,116]. Define the margin of a hypothesis f at a labeled sample $(\mathbf{x}, y)$ as:

$$\rho_f(\mathbf{x}, y) \triangleq \frac{1}{2}(f(\mathbf{x}, y) - \max_{y' \neq y} f(\mathbf{x}, y')) \tag{8.7}$$

Then the margin loss of a hypothesis f in the distribution D can be defined as:

$$err_D^{\rho}(f) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim D} \Phi_\rho \circ \rho_f(\mathbf{x}, y) \tag{8.8}$$

where $\circ$ denotes function composition and Φ_ρ is defined as:

$$\Phi_\rho(\mathbf{x}) \triangleq \begin{cases} 0 & \rho \leq \mathbf{x} \\ 1 - \mathbf{x}/\rho & 0 \leq \mathbf{x} \leq \rho \\ 1 & \mathbf{x} \leq 0 \end{cases} \tag{8.9}$$

Given $h, h' \in \mathcal{H}$, the expected *0-1* disparity between h and h' in the distribution D can be defined as:

$$dis_D(h, h') \triangleq \mathbb{E}_D \mathbb{1}[h' \neq h] \tag{8.10}$$

Furthermore, given $h \in \mathcal{H}$, the *0-1* discrepancy induced by $h' \in \mathcal{H}$ between two dissimilar distributions D_1 and D_2 can be defined as:

$$\begin{aligned} d_{h,\mathcal{H}}(D_1, D_2) &\triangleq \sup_{h' \in \mathcal{H}} (dis_{D_2}(h, h') - dis_{D_1}(h, h')) \\ &= \sup_{h' \in \mathcal{H}} (\mathbb{E}_{D_2} \mathbb{1}[h' \neq h] - \mathbb{E}_{D_1} \mathbb{1}[h' \neq h]) \end{aligned} \tag{8.11}$$

If we use the margin loss to replace the *0-1* loss, the margin disparity is written as:

$$dis_D^{\rho}(f, f') \triangleq \mathbb{E}_{\mathbf{x} \sim D} \Phi_{\rho} \circ \rho_{f'}(\mathbf{x}, h_f(\mathbf{x})) \tag{8.12}$$

Where f and f' are scoring functions while h_f and $h_{f'}$ are their corresponding labeling functions. The margin discrepancy can be formularized based on the margin disparity:

$$d_{f,\mathcal{F}}^{\rho}(D_1, D_2) \triangleq \sup_{f' \in \mathcal{F}} (dis_{D_2}^{\rho}(f, f') - dis_{D_1}^{\rho}(f, f')) \tag{8.13}$$

The margin discrepancy is a well-proved measurement to gauge the domain divergence in domain adaptation [113].

8.3 Methodology

In this section, we first theoretically analyze how to incorporate the margin discrepancy into MDTC and derive the generalization bound for MDTC based on the margin discrepancy. Second, according to the above theoretical finding, we propose a margin discrepancy-based method for MDTC. We will provide the theoretical proof details in Section 8.3.2.

The MDAT method is mainly inspired by three point: (1) The margin discrepancy

has never been applied in multi-domain text classification; (2) To date, no generalization bound is induced for MDTC; (3) There are strong connections between multi-domain text classification and domain adaptation, the MDTC task can be separated into M domain adaptation tasks such that we could extend the domain adaptation theories [26] to multi-domain scenarios.

8.3.1 Theoretical Analysis

The margin discrepancy is a well-proved discrepancy that can efficiently measure the distribution distance between two domains in the shared latent space for domain adaptation due to the following proposition [113]:

Proposition 8.1. *For any scoring function $f \in \mathcal{F}$,*

$$err_{D_T}(f) \leq err^{\rho}_{D_S}(f) + d^{\rho}_{f,\mathcal{F}}(D_T, D_S) + \lambda \tag{8.14}$$

where λ is a constant which is independent of f.

In equation 8.14, $err^{\rho}_{D_S}(f)$ means the performance of f on the source domain D_S and the margin discrepancy $d^{\rho}_{f,\mathcal{F}}(D_T, D_S)$ bounds the performance gap caused by the distribution shift between the source domain D_S and the target domain D_T. λ is a constant determined by the practical problem. As the MDTC task can be divided into M domain adaptation tasks between $\{D_i\}_{i=1}^{M}$ and $\bar{D}$, where $\bar{D} = \frac{\sum_{i=1}^{M} D_i}{M}$. By taking the M domain adaptation tasks into consideration, we intuitively extend equation 8.14 as:

$$err_{\bar{D}}(f) \leq \frac{1}{M} \sum_{i=1}^{M} [err^{\rho}_{D_i}(f) + d^{\rho}_{f,\mathcal{F}}(D_i, \bar{D})] + \lambda \tag{8.15}$$

The updated learning bound provides theoretical guarantees to advance MDTC by optimizing the domain divergence between each D_i and $\bar{D}$, and gives new perspectives

for analyzing MDTC theoretically. To induce the generalization bound for MDTC, we first present the basic notations of the Rademacher complexity. The Rademacher complexity is widely used for deriving generalization bound in machine learning, it is a measurement of richness for a particular hypothesis space [114].

Definition 8.1. *(Rademacher Complexity) Assume $\mathcal{F}$ is the hypothesis space which maps $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to [a,b], and $\widehat{D} = \{z_1, z_2, ..., z_n\}$ denotes a dataset of size n whose samples are drawn from the distribution D over $\mathcal{Z}$. Then the empirical Rademacher complexity of $\mathcal{F}$ with regard to the dataset $\widehat{D}$ is defined as:*

$$\widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{F}) \triangleq \mathbb{E}_{\sigma} \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(z_i) \tag{8.16}$$

where σ_i's are independent uniform random variables taking values in $\{-1, +1\}$. The Rademacher complexity is:

$$\mathfrak{R}_{n,D}(\mathcal{F}) \triangleq \mathbb{E}_{\widehat{D} \sim D} \widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{F}) \tag{8.17}$$

Then, the margin discrepancy between the two distributions D_1 and D_2 could be approximated through finite samples using the Rademacher complexity.

Lemma 8.1. *For any $\delta > 0$, with probability $1 - 2\delta$,*

$$\begin{aligned} |d^{\rho}_{f,\mathcal{F}}(\widehat{D}_1, \widehat{D}_2) - d^{\rho}_{f,\mathcal{F}}(D_1, D_2)| &\leq \frac{k}{\rho} \mathfrak{R}_{n_1, D_1}(\Pi_{\mathcal{H}} \mathcal{F}) \\ + \frac{k}{\rho} \mathfrak{R}_{n_2, D_2}(\Pi_{\mathcal{H}} \mathcal{F}) &+ \sqrt{\frac{\log \frac{2}{\delta}}{2n_1}} + \sqrt{\frac{\log \frac{2}{\delta}}{2n_2}} \end{aligned} \tag{8.18}$$

Where $\Pi_{\mathcal{H}} \mathcal{F} = \{\mathbf{x} \mapsto f(\mathbf{x}, h_f(\mathbf{x})) | h_f \in \mathcal{H}, f \in \mathcal{F}\}$ is the scoring version of the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$ [113], n_1 and n_2 are sizes of datasets $\widehat{D}_1$ and $\widehat{D}_2$, respectively. If we combine proposition 8.1 with lemma 8.1, a new Rademacher complexity generalization bound for MDTC can be derived:

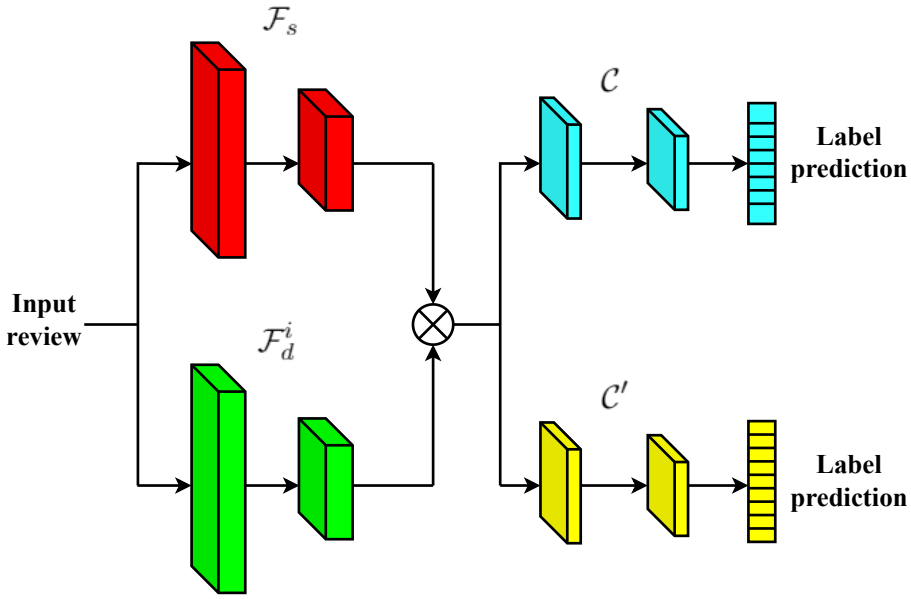


Figure 12: The architecture of the MDAT method.

Theorem 8.1. *(Generalization Bound) For all $\delta > 0$, with probability $1 - 3\delta$,*

$$
\begin{aligned}
err_{\bar{D}}(f) \leq & \frac{1}{M}\sum_{i=1}^{M}[err^{\rho}_{\widehat{D}_i}(f) + d^{\rho}_{f,\mathcal{F}}(D_i, \bar{D})] + \lambda \\
& + \frac{1}{M}\sum_{i=1}^{M}[\frac{8}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_1\mathcal{F}) + \frac{2}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_{\mathcal{H}}\mathcal{F}) \\
& + 2\sqrt{\frac{\log\frac{2}{\delta}}{2n_i}}] + \frac{2}{\rho}\mathfrak{R}_{\bar{n},\bar{D}}(\Pi_{\mathcal{H}}\mathcal{F}) + \sqrt{\frac{\log\frac{2}{\delta}}{2\bar{n}}}
\end{aligned} \tag{8.19}
$$

Where $\bar{n} = \frac{\sum_{i=1}^{M} n_i}{M}$. In summary, this work tries to fill the gap between the MDTC algorithms and theories. We here provide the generalization bound for MDTC, which supports the MDTC algorithms from the theoretical perspective. Compared with the convergence condition proposed in [10], our work is more informative.

8.3.2 Theoretical Proof

Theorem 8.2. *(Proposition 3.3 of [113] & Proposition 8.1 in Section 8.3.1) For any scoring function $f \in \mathcal{F}$,*

$$err_{D_T}(f) \leq err^{\rho}_{D_S}(f) + d^{\rho}_{f,\mathcal{F}}(D_T, D_S) + \lambda$$

where $\lambda = \lambda(\rho, \mathcal{F}, D_S, D_T)$ *is a constant independent of* f*.*

Definition 8.2. *(Definition 3.4 of [113]) Given a hypothesis space* $\mathcal{F}$ *and its corresponding labeling space* $\mathcal{H}$*,* $\Pi_{\mathcal{H}}\mathcal{F}$ *is defined as*

$$\Pi_{\mathcal{H}}\mathcal{F} = \{x \mapsto f(x, h_f(x)) | h_f \in \mathcal{H}, f \in \mathcal{F}\}$$

[117] presents a geometric interpretation of the set $\Pi_{\mathcal{H}}\mathcal{F}$. Assume that $\mathcal{X}$ is a manifold, we can obtain a vector bundle $\mathcal{B}$ by assigning a vector space $\mathbb{R}^k$ to each sample in $\mathcal{X}$. By considering the values of $\mathcal{H}$ as one-hot vectors in $\mathbb{R}^k$, $\mathcal{H}$ and $\mathcal{F}$ are both sets of sections of $\mathcal{B}$ containing vector fields. $\Pi_{\mathcal{H}}\mathcal{F}$ can be treated as the space of inner products of vector fields with $\mathcal{H}$ and $\mathcal{F}$:

$$\Pi_{\mathcal{H}}\mathcal{F} = \langle \mathcal{H}, \mathcal{F} \rangle = \{\langle h, f \rangle | h \in \mathcal{H}, f \in \mathcal{F}\}$$

Lemma 8.2. *(A modified version of Theorem 8.1, [114]). Suppose* $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$ *is the hypothesis set of scoring functions with* $\mathcal{Y} = \{1, 2, ..., k\}$*. Let*

$$\Pi_1\mathcal{F} \triangleq \{x \mapsto f(x, y) | y \in \mathcal{Y}, f \in \mathcal{F}\}$$

Fix $\rho > 0$*. Then for any* $\delta > 0$*, with probability at least* $1 - \delta$*, the following holds for all* $f \in \mathcal{F}$*:*

$$|err^{\rho}_{D}(f) - err^{\rho}_{\widehat{D}}(f)| \leq \frac{2k^2}{\rho}\mathfrak{R}_{n,D}(\Pi_1\mathcal{F}) + \sqrt{\frac{\log \frac{2}{\delta}}{2n}}$$

A simple corollary of this lemma is the margin bound for binary classification.

$$err_D(f) \leq err^{\rho}_{D}(f) \leq err^{\rho}_{\widehat{D}}(f) + \frac{8}{\rho}\mathfrak{R}_{n,D}(\Pi_1\mathcal{F}) + \sqrt{\frac{\log \frac{2}{\delta}}{2n}} \tag{8.20}$$

Lemma 8.3. *(Talagrand's lemma, [118]; [114]). Let $\Phi : \mathbb{R} \mapsto \mathbb{R}$ be an ℓ-Lipschitz. Then for any hypothesis space $\mathcal{F}$ of real-valued functions, and any sample D of size n, the following inequality holds:*

$$\mathfrak{R}_{\widehat{D}}(\Phi \circ \mathcal{F}) \leq \ell \mathfrak{R}_{\widehat{D}}(\mathcal{F})$$

Lemma 8.4. *(Lemma 8.1 of [114]). Let $\mathcal{F}_1, ..., \mathcal{F}_k$ be k hypothesis sets in $\mathbb{R}^{\mathcal{X}}$, $k > 1$. $\mathcal{G} = \{\max\{f_1, ..., f_k\} : f_i \in \mathcal{F}, i \in \{1, ..., k\}\}$, then for any dataset $\widehat{D}$ of size n, we have:*

$$\widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{G}) \leq \sum_{i=1}^{k} \widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{F}_i)$$

Theorem 8.3. *(Lemma 8.1 in the Section 8.3.1). Let $\mathcal{F} \in \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$ is a hypothesis space. Let $\mathcal{H}$ be the set of classifiers (mapping $\mathcal{X}$ to $\mathcal{Y}$) corresponding to $\mathcal{F}$. For any $\delta > 0$, with probability $1 - 2\delta$, the following holds simultaneously for any scoring function f,*

$$|d^{\rho}_{f,\mathcal{F}}(\widehat{D}_1, \widehat{D}_2) - d^{\rho}_{f,\mathcal{F}}(D_1, D_2)| \leq \frac{k}{\rho}\mathfrak{R}_{n_1,D_1}(\Pi_{\mathcal{H}}\mathcal{F}) + \frac{k}{\rho}\mathfrak{R}_{n_2,D_2}(\Pi_{\mathcal{H}}\mathcal{F}) + \sqrt{\frac{\log\frac{2}{\delta}}{2n_1}} + \sqrt{\frac{\log\frac{2}{\delta}}{2n_2}}$$

Proof. For $\forall f, f' \in \mathcal{F}$, define the τ_f-transform of f' to be

$$\tau_f f'(x, y) = \begin{cases} f'(x, 1) & \text{if} \quad y = h_f(x) \\ f'(x, h_f(x)) & \text{if} \quad y = 1 \\ f'(x, y) & \text{else} \end{cases}$$

Where h_f is the induced classifier mapping from $\mathcal{X}$ to $\mathcal{Y}$. Let $\mathcal{G} = \{\tau_f f' | f, f' \in \mathcal{F}\}$, $\widetilde{\mathcal{G}} = \{(x, y) \mapsto \rho_g(x, y) | g \in \mathcal{G}\}$

Consider the family of functions $\Phi_\rho \circ \widetilde{\mathcal{G}}$ which takes values in $[0, 1]$. Through

lemma 8.2, with probability at least $1-\delta$, for $\forall g \in \mathcal{G}$.

$$\begin{aligned} |err_D^\rho(g) - err_{\widehat{D}}^\rho(g)| &= |\mathbb{E}\Phi_\rho \circ \rho_g(x,y) - \frac{1}{n}\sum_{i=1}^{n} \Phi_\rho \circ \rho_g(x_i,y_i)| \\ &\leq 2\mathfrak{R}_{n,D}(\Phi_\rho \circ \widetilde{\mathcal{G}}) + \sqrt{\frac{\log\frac{2}{\delta}}{2n}} \end{aligned} \tag{8.21}$$

Regard all the data as from the same class 1. Define:

$$\mathfrak{R}^0_{n,D}(\mathcal{G}) = \mathbb{E}_{(x_i,1),x_i\sim D}\widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{G})$$

Then we can update equation 8.21 to:

$$\begin{aligned} |dis_D^\rho(g,1) - disp_{\widehat{D}}^\rho(g,1)| &= |\mathbb{E}\Phi_\rho \circ \rho_g(x,1) - \frac{1}{n}\sum_{i=1}^{n} \Phi_\rho \circ \rho_g(x_i,1)| \\ &\leq 2\mathfrak{R}^0_{n,D}(\Phi_\rho \circ \widetilde{\mathcal{G}}) + \sqrt{\frac{\log\frac{2}{\delta}}{2n}} \end{aligned}$$

For any $f, f' \in \mathcal{F}$, let $g = \tau_f f'$. Then $g \in \mathcal{G}$ and $dis_D^\rho(g,1) = dis_D^\rho(f',f)$, $dis_{\widehat{D}}^\rho(g,1) = dis_{\widehat{D}}^\rho(f',f)$ Thus, we can obtain:

$$\begin{aligned} &\sup_{f,f'\in\mathcal{F}} |dis_D^\rho(f',f) - dis_{\widehat{D}}^\rho(f',f)| \leq \sup_{g\in\mathcal{G}} |dis_D^\rho(g,1) - dis_{\widehat{D}}^\rho(g,1)| \\ &\leq 2\mathfrak{R}^0_{n,D}(\Phi_\rho \circ \widetilde{\mathcal{G}}) + \sqrt{\frac{\log\frac{2}{\delta}}{2n}} \end{aligned}$$

Through Lemma 8.3, $\mathfrak{R}^0_{n,D}(\Phi_\rho \circ \widetilde{\mathcal{G}}) \leq \frac{1}{\rho}\mathfrak{R}^0_{n,D}(\widetilde{\mathcal{G}})$

$$\begin{aligned} \mathfrak{R}^0_{n,D}(\widetilde{\mathcal{G}}) =& \frac{1}{n}\mathbb{E}_{S,\sigma}(\sup_{g\in\mathcal{G}}\sum_{i=1}^{n}\sigma_i\rho_g(x_i,1)) \\ =& \frac{1}{n}\mathbb{E}_{S,\sigma}(\sup_{g\in\mathcal{G}}\sum_{i=1}^{n}\sigma_i(g(x_i,1) - \max_{y\neq 1} g(x_i,y))) \\ =& \frac{1}{n}\mathbb{E}_{S,\sigma}(\sup_{f,f'\in\mathcal{F}}\sum_{i=1}^{n}\sigma_i(f'(x_i,h_f(x_i))) - \max_{y\neq h_f(x_i)} f'(x_i,y)) \\ \leq& \frac{1}{n}\mathbb{E}_{S,\sigma}\sup_{f\in\mathcal{F},h\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i f(x_i,h(x_i)) + \frac{1}{n}\mathbb{E}_{S,\sigma}\sup_{f\in\mathcal{F},h\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i(-\max_{y\neq h(x_i)} f'(x_i,y)) \\ =& \mathfrak{R}_{n,D}(\Pi_{\mathcal{H}}\mathcal{F}) + \frac{1}{n}\mathbb{E}_{S,\sigma}\sup_{f\in\mathcal{F},h\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i \max_{y\neq h(x_i)} f(x_i,y) \end{aligned}$$

Define the permutation

$$\xi(i) = \begin{cases} i+1 & i = 1, ..., k-1 \\ \\ 1 & i = k \end{cases}$$

By our assumption of $\mathcal{H}$, we have the result that $\forall h \in \mathcal{H}$, $\xi^j h \in \mathcal{H}, j = 1, 2, ..., k-1$.

$$\frac{1}{n}\mathbb{E}_{S,\sigma} \sup_{f\in\mathcal{F}, h\in\mathcal{H}} \sum_{i=1}^{n} \sigma_i \max_{y\neq h(x_i)} f(x_i, y) = \frac{1}{n}\mathbb{E}_{S,\sigma} \sup_{f\in\mathcal{F}, h\in\mathcal{H}} \sum_{i=1}^{n} \sigma_i \max_{j\in\{1,...,k-1\}} f(x_i, \xi^j h(x_i))$$

Let $\Pi_{\mathcal{H}}\mathcal{F}^{(k-1)} = \{max\{f_1, ..., f_{k-1}\} | f_i \in \Pi_{\mathcal{H}}\mathcal{F}, i = 1, ..., k-1\}$. Then applying Lemma 8.4, we can obtain:

$$\begin{aligned} \frac{1}{n}\mathbb{E}_{S,\sigma} \sup_{f,h} \sum_{i=1}^{n} \sigma_i \max_{j\in\{1,...,k-1\}} f(x_i, \xi^j h(x_i)) &= \frac{1}{n}\mathbb{E}_{S,\sigma} \sup_{f\in\Pi_{\mathcal{H}}\mathcal{F}^{(k-1)}} \sum_{i=1}^{n} \sigma_i f(x_i) \\ &\leq \frac{k-1}{n}\mathbb{E}_{S,\sigma} \sup_{f\in\Pi_{\mathcal{H}}\mathcal{F}} \sum_{i=1}^{n} \sigma_i f(x_i) \end{aligned}$$

Therefore, we have

$$\begin{aligned} \mathfrak{R}^0_{n,D}(\widetilde{\mathcal{G}}) &\leq \mathfrak{R}_{n,D}(\Pi_{\mathcal{H}}\mathcal{F}) + \frac{k-1}{n}\mathbb{E}_{S,\sigma} \sup_{f\in\Pi_{\mathcal{H}}\mathcal{F}} \sum_{i=1}^{n} \sigma_i f(x_i) \\ &\leq k\mathfrak{R}_{n,D}(\Pi_{\mathcal{H}}\mathcal{F}), \\ &\sup_{f,f'\in\mathcal{F}} |dis^{\rho}_{D}(f', f) - dis^{\rho}_{\widehat{D}}(f', f)| \\ &\leq \frac{2k}{\rho}\mathfrak{R}_{n,D}(\Pi_{\mathcal{H}}\mathcal{F}) + \sqrt{\frac{\log\frac{2}{\delta}}{2n}} \end{aligned}$$

Similarly

$$\sup_{f,f'\in\mathcal{F}} |dis^{\rho}_{D_i}(f', f) - dis^{\rho}_{\widehat{D}_i}(f', f)| \leq \frac{2k}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_{\mathcal{H}}\mathcal{F}) + \sqrt{\frac{\log\frac{2}{\delta}}{2n_i}}$$

Therefore, we conclude

$$\begin{aligned}\sup_{f\in\mathcal{F}}|d^{\rho}_{f,\mathcal{F}}(\widehat{D}_1,\widehat{D}_2)-d^{\rho}_{f,\mathcal{F}}(D_1,D_2)| \leq & \sup_{f,f'\in\mathcal{F}}|dis^{\rho}_{D_2}(f',f)-dis^{\rho}_{\widehat{D}_2}(f',f)| \\ & + \sup_{f,f'\in\mathcal{F}}|dis^{\rho}_{D_1}(f',f)-dis^{\rho}_{\widehat{D}_1}(f',f)| \\ \leq & \frac{2k}{\rho}\mathfrak{R}_{n_1,D_1}(\Pi_{\mathcal{H}}\mathcal{F})+\frac{2k}{\rho}\mathfrak{R}_{n_2,D_2}(\Pi_{\mathcal{H}}\mathcal{F})+\sqrt{\frac{\log\frac{2}{\delta}}{2n_1}}+\sqrt{\frac{\log\frac{2}{\delta}}{2n_2}}\end{aligned}$$

□

Theorem 8.4. *(Theorem 8.1 in Section 8.3.1). For any $\delta>0$, with probability $1-3\delta$, we have the following uniform generalization bound for all scoring functions f:*

$$\begin{aligned}err_{\bar{D}}(f) \leq & \frac{1}{M}\sum_{i=1}^{M}[err^{\rho}_{\widehat{D}_i}(f)+d^{\rho}_{f,\mathcal{F}}(D_i,\bar{D})]+\lambda+\frac{1}{M}\sum_{i=1}^{M}[\frac{8}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_1\mathcal{F})+\frac{2}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_{\mathcal{H}}\mathcal{F}) \\ & +2\sqrt{\frac{\log\frac{2}{\delta}}{2n_i}}]+\frac{2}{\rho}\mathfrak{R}_{\bar{n},\bar{D}}(\Pi_{\mathcal{H}}\mathcal{F})+\sqrt{\frac{\log\frac{2}{\delta}}{2\bar{n}}}\end{aligned}$$

Before obtaining this theorem, we combine Theorem 8.2, Equation 8.20 *and Theorem 8.3 to get:*

$$\begin{aligned}err_{\bar{D}}(f) \leq & err^{\rho}_{\widehat{D}_i}(f)+d^{\rho}_{f,\mathcal{F}}(\widehat{D}_i,\widehat{\bar{D}})+\lambda+\frac{8}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_1\mathcal{F})+\frac{2}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_{\mathcal{H}}\mathcal{F}) \\ & +2\sqrt{\frac{\log\frac{2}{\delta}}{2n_i}}+\frac{2}{\rho}\mathfrak{R}_{\bar{n},\bar{D}}(\Pi_{\mathcal{H}}\mathcal{F})+\sqrt{\frac{\log\frac{2}{\delta}}{2\bar{n}}}\end{aligned}$$

By simple summation, we can get

$$\begin{aligned}M\times err_{\bar{D}}(f) \leq & \sum_{i=1}^{M}[err^{\rho}_{\widehat{D}_i}(f)+d^{\rho}_{f,\mathcal{F}}(\widehat{D}_i,\widehat{\bar{D}})+\lambda+\frac{8}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_1\mathcal{F}) \\ & +\frac{2}{\rho}\mathfrak{R}_{n_i,D_i}(\Pi_{\mathcal{H}}\mathcal{F})+2\sqrt{\frac{\log\frac{2}{\delta}}{2n_i}}+\frac{2}{\rho}\mathfrak{R}_{\bar{n},\bar{D}}(\Pi_{\mathcal{H}}\mathcal{F})+\sqrt{\frac{\log\frac{2}{\delta}}{2\bar{n}}}]\end{aligned}$$

Then we can conclude the Theorem 8.4

8.3.3 Margin Discrepancy-based Adversarial Training

According to the above theoretical analysis, we propose the margin discrepancy-based adversarial training (MDAT) method for MDTC. As the margin discrepancy measures the domain divergence by restricting the search range in the hypothesis space, the proposed MDAT method does not need the domain discriminator to implement the adversarial training. We here adopt the classifier-based adversarial training paradigm [49]. There are four components in our MDAT model: a shared feature extractor $\mathcal{F}_s$, M domain-specific feature extractors $\{\mathcal{F}_d^i\}_{i=1}^M$, a classifier $\mathcal{C}$ and an auxiliary classifier $\mathcal{C}'$, the architecture is shown in Figure 12. In contrast with the standard MDAT method, such as multinomial adversarial network [10], we do not use the domain discriminator and introduce the auxiliary classifier. The shared feature extractor $\mathcal{F}_s$ aims to learn domain-invariant features that can contribute to the classification tasks across all domains. The domain-specific feature extractor $\mathcal{F}_d^i$ seeks to capture domain-specific features from the i-th domain $\widehat{D}_i$ that only benefit the classification task on $\widehat{D}_i$. The auxiliary classifier $\mathcal{C}'$ helps the main classifier $\mathcal{C}$ to perform the minimax optimization against the shared feature extractor $\mathcal{F}_s$. The main classifier $\mathcal{C}$ is used to conduct the text classification. The output of a (shared/domain-specific) feature extractor is a fixed-length vector. The classifiers take the concatenation of a domain-invariant feature vector and a domain-specific feature vector as input and output the label probability.

According to Theorem 8.1, we aim to minimize the classification errors across all datasets $\{\widehat{D}_i\}_{i=1}^M$ and the total f-divergence between each of the distributions of the M domains and the centroid of the M distributions. Therefore, we formularize the MDTC task into the following minimization problem in the hypothesis space $\mathcal{F}$:

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{M} [err_{D_i}^{\rho}(f) + d_{f,\mathcal{F}}^{\rho}(D_i, \bar{D})] \tag{8.21}$$

As demonstrated in [113], minimizing the margin discrepancy is equivalent to performing a minimax optimization since the margin discrepancy is defined as the supremum over the hypothesis space $\mathcal{F}$. If we denote the feature embedding function as ϕ and the classifying function as ω, the scoring function f can be decomposed as $f = \omega \circ \phi$. Applying ϕ and ω to the M domains, the overall optimization problem can be defined as follows:

$$\begin{aligned} \min_{\omega,\phi} \sum_{i=1}^{M} [err^{\rho}_{\phi(\widehat{D}_i)}(\omega) + dis^{\rho}_{\phi(\widehat{\bar{D}})}(\omega, \omega^*) - dis^{\rho}_{\phi(\widehat{D}_i)}(\omega, \omega^*)] \\ \omega^* = \max_{\omega'} \sum_{i=1}^{M} [dis^{\rho}_{\phi(\widehat{\bar{D}})}(\omega, \omega') - dis^{\rho}_{\phi(\widehat{D}_i)}(\omega, \omega')] \end{aligned} \tag{8.22}$$

Where ω' is the classifying function of the auxiliary classifier $\mathcal{C}'$. By adversarially implementing the minimax optimization on the outputs of the two classifiers, the domain-invariance can be enforced to the shared latent space.

As the margin loss can cause gradient vanishing in stochastic gradient descent in practice [116], and thus cannot be optimized efficiently for representation learning that heavily relies on gradient propagation, we need to use other loss functions to approximate the margin discrepancy in mini-batch stochastic gradient descent. Here we use the cross-entropy loss to reformulate the above optimization.

$$\min_{\mathcal{F}_s, \mathcal{F}_d^i, \mathcal{C}} \max_{\mathcal{C}'} \mathcal{J}_C + \alpha \mathcal{J}_D \tag{8.23}$$

$$\mathcal{J}_C = -\sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{L}_i} \log(\mathcal{C}_y[\mathcal{F}^i(\mathbf{x})]) \tag{8.24}$$

$$\begin{aligned} \mathcal{J}_D = \sum_{i=1}^{M} [\beta \mathbb{E}_{\mathbf{x}\sim\mathbb{L}_i\cup\mathbb{U}_i} \log(\mathcal{C}'_{\sigma[\mathcal{F}^i(\mathbf{x})]}[\mathcal{F}^i(\mathbf{x})]) \\ + \mathbb{E}_{\mathbf{x}\sim\mathbb{L}_i\cup\mathbb{U}_i} \log(1 - \mathcal{C}'_{\sigma[\mathcal{F}^i(\mathbf{x})]}[\mathcal{F}^i(\mathbf{x})])] \end{aligned} \tag{8.25}$$

Where $\beta = \exp(\rho)$ is supposed to attain the margin, $\mathcal{F}^i(\mathbf{x}) = [\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^i(\mathbf{x})]$ is the input of the classifiers where $[\cdot, \cdot]$ is the concatenation of two vectors. $\mathcal{C}_y([\mathcal{F}^i(\mathbf{x})])$ and $\mathcal{C}'_y([\mathcal{F}^i(\mathbf{x})])$ indicate the prediction probability of the given sample $\mathbf{x}$ belonging to the label y by the classifier $\mathcal{C}$ and $\mathcal{C}'$, respectively. $\sigma(\cdot)$ denotes the labelling function of the classifier $\mathcal{C}$. α is the hyperparameter used to balance two competitive terms. Through implementing the adversarial training, our proposed MDAT approach can reach the optimum where $\phi(\widehat{D}_1) \approx \phi(\widehat{D}_2) \approx ... \approx \phi(\widehat{D}_M) \approx \phi(\widehat{\widehat{D}})$.

8.3.4 Training procedure

We use the mini-batch stochastic gradient descent and the alternating training strategy [21] to train the MDAT approach. There exist two data flows in the training, the input data are entered into both the shared feature extractor and the corresponding domain-specific feature extractor to obtain domain-invariant and domain-specific features, respectively. The concatenation of a domain-invariant feature vector and a domain-specific feature vector is used as the input of the classifiers. The shared feature extractor is trained against the two classifiers to conduct the adversarial training to generate the domain-invariant features. The training procedure can be divided into three steps: (1) Calculate $l_C = \mathcal{J}_C$ on the labeled data and update the parameters of $\mathcal{F}_s$, $\mathcal{F}_d^i$, $\mathcal{C}$ and $\mathcal{C}'$ by descending along gradients ∇l_C; (2) Calculate $loss = \mathcal{J}_C + \alpha \mathcal{J}_D$ on both the labeled and unlabeled data, fix the parameters of $\mathcal{C}$ and $\mathcal{C}'$, and update the parameters of $\mathcal{F}_s$ and $\mathcal{F}_d^i$ by descending along gradients $\nabla loss$; (3) Calculate $l_D = \mathcal{J}_D$ on both the labeled and unlabeled data, fix the parameters of $\mathcal{F}_s$ and $\mathcal{F}_d^i$, and update the parameters of $\mathcal{C}$ and $\mathcal{C}'$ by ascending along the descents ∇l_D. There are two hyperparameters in the MDAT approach: α and β. α is used to balance different loss functions and β is used to control the margin. In our experiments, we fix $\alpha = 0.5$ and $\beta = 4$. The detailed algorithm is presented in Algorithm 5.

Algorithm 5 SGD training algorithm

1: **Input:** labeled data $\mathbb{L}_i$ and unlabeled data $\mathbb{U}_i$ in M domains; hyperparameters: α and β

2: **for** number of training iterations **do**

3: Sample labeled mini-batches from the multiple domains $B^\ell = \{B_1^\ell, \cdots, B_M^\ell\}$.

4: Sample unlabeled mini-batches from the multiple domains $B^u = \{B_1^u, \cdots, B_M^u\}$.

5: Calculate $l_C = \mathcal{J}_C$ on B^ℓ;

Update $\mathcal{F}_s$, $\mathcal{F}_d^i$, $\mathcal{C}$ and $\mathcal{C}'$ by descending along gradients ∇l_C.

6: Calculate $loss = \mathcal{J}_C + \alpha \mathcal{J}_D$ on B^ℓ and B^u;

Fix $\mathcal{C}$ and $\mathcal{C}'$, and update $\mathcal{F}_s$ and $\mathcal{F}_d^i$ by descending along gradients $\nabla loss$.

7: Calculate $l_D = \mathcal{J}_D$ on B^ℓ and B^u;

Fix $\mathcal{F}_s$ and $\mathcal{F}_d^i$, and update $\mathcal{C}$ and $\mathcal{C}'$ by ascending along gradients ∇l_D.

8: **end for**

Table 15: The classification accuracies of MDAT on the Amazon review dataset.

Domain	CMSC-LS	CMSC-SVM	CMSC-Log	MAN-L2	MAN-NLL	DACL	MRAN	CAN	CRAL	MDAT
Books	82.10	82.26	81.81	82.46	82.98	83.45	84.60	83.76	**85.26**	84.33 ± 0.16
DVD	82.40	83.48	83.73	83.98	84.03	85.50	85.60	84.68	85.83	$\mathbf{86.07 \pm 0.11}$
Electr.	86.12	86.76	86.67	87.22	87.06	87.40	89.10	88.34	**89.32**	88.74 ± 0.06
Kit.	87.56	88.20	88.23	88.53	88.57	90.00	91.25	90.03	**91.60**	90.56 ± 0.13
AVG	84.55	85.18	85.11	85.55	85.66	86.59	87.64	86.70	**88.00**	87.43 ± 0.08

8.4 Experiments

In this section, we first present the implementation details. Second, we list the baselines used in the comparison. Third, we show the MDTC experimental results. Lastly, we provide extensive analyses on ablation study, parameter sensitivity analysis, and multi-source unsupervised domain adaptation to give more insights into our MDAT approach.

8.4.1 Implementation Details

We implemented our approach using Pytorch. The proposed MDAT approach has two hyperparameters: α and β, we fix $\alpha = 0.5$ and $\beta = 4$. In the experiments, we set the domain-invariant feature dimension as 128 and the domain-specific feature dimension as 64. The classifiers $\mathcal{C}$ and $\mathcal{C}'$ are both MLPs whose input dimensions are 128+64 and 128+64, respectively. They both have one hidden layer whose dimension is the same as the input dimension. We use ReLU as the activation function in each component. The dropout rate for each component is set as 0.4. The Adam optimizer [86] with a learning rate of 0.0001 is used to train the model. We set the batch size as 8 and the number of training iterations as 50. For the experiments on the Amazon review dataset, MLPs with two hidden layers are used as the feature extractors, the two hidden layers contain 1,000 and 500 units respectively. The input dimension of

Table 16: The classification accuracies of MDAT on the FDU-MTL dataset.

Domain	MT-CNN	MT-DNN	ASP-MTL	MAN-L2	MAN-NLL	DACL	MRAN	CAN	CRAL	MDAT
books	84.5	82.2	84.0	87.6	86.8	87.5	87.0	87.8	**89.3**	88.1 ± 0.2
electronics	83.2	88.3	86.8	87.4	88.8	90.3	89.0	**91.6**	89.1	88.8±0.6
dvd	84.0	84.2	85.5	88.1	88.6	89.8	89.0	89.5	**91.0**	**91.0 ± 0.4**
kitchen	83.2	80.7	86.2	89.8	89.9	91.5	**93.0**	90.8	92.3	92.1 ± 0.4
apparel	83.7	85.0	87.0	87.6	87.6	89.5	91.5	87.0	**91.6**	90.3 ± 0.5
camera	86.0	86.2	89.2	91.4	90.7	91.5	93.0	93.5	**96.3**	92.2 ± 0.4
health	87.2	85.7	88.2	89.8	89.4	**90.5**	90.0	90.4	87.8	89.8±0.2
music	83.7	84.7	82.5	85.9	85.5	86.3	86.5	86.9	**88.1**	87.5 ± 0.1
toys	89.2	87.7	88.0	90.0	90.4	91.3	86.0	90.0	**91.6**	91.4 ± 0.4
video	81.5	85.0	84.5	89.5	89.6	88.5	88.5	88.8	**92.6**	90.7 ± 0.6
baby	87.7	88.0	88.2	90.0	90.2	**92.0**	90.0	**92.0**	90.9	90.0±0.3
magazine	87.7	89.5	92.2	92.5	92.9	93.8	93.5	94.5	**95.2**	94.2 ± 0.2
software	86.5	85.7	87.2	90.4	90.9	90.5	89.5	90.9	87.7	**91.3 ± 0.5**
sports	84.0	83.2	85.7	89.0	89.0	89.3	90.5	91.2	**91.3**	89.7 ± 0.3
IMDb	86.2	83.2	85.5	86.6	87.0	87.3	89.0	88.5	**90.8**	89.4 ± 0.1
MR	74.5	75.5	76.7	76.1	76.7	76.0	**78.5**	77.1	77.3	77.0 ± 0.5
AVG	84.5	84.3	86.1	88.2	88.4	89.1	89.0	89.4	**90.2**	89.6 ± 0.1

feature extractors is 5,000. For the experiments on the FDU-MTL dataset, CNN with one convolutional layer is used as the feature extractor. Its kernel sizes are (3, 4, 5), and the number of kernels is 200. The input dimension of the convolutional layer is 100, we use the word embedding, obtained by processing each word of the input sequence through word2vec [87], as the input of the feature extractors.

8.4.2 Comparison Methods

We compare our MRAN model with several state-of-the-art baselines:

- CMSC-LS, CMSC-SVM, CMSC-Log: The collaborative multi-domain sentiment classification method which trains a shareable classifier and a set of domain-specific classifiers for all domains. The final prediction is determined by the combination of the outputs of the shareable classifier and the corresponding classifier of the given instance. The CMSC model is trained with squared loss, hinge loss, and log loss respectively [85].

- MAN: The multinomial adversarial network for multi-domain text classification [10]. This method uses two forms of loss functions to train the multinomial domain discriminator: The negative log-likelihood loss (MAN-NLL) and the least square loss (MAN-L2). The MAN also induces the convergence conditional for adversarial MDTC with regard to the two loss functions.

- MT-CNN: The deep neural network model learns features across different tasks given very limited prior knowledge by using a single convolutional layer [58].

- MT-DNN: The multi-task deep neural network model with bag-of-words input and MLPs [89], in which a hidden layer is shared.

- ASP-MTL: The adversarial multi-task learning model for text classification [13], in which shared-private paradigm, adversarial training, and an orthogonal restriction are used to guide feature extraction.

- DACL: The dual adversarial co-learning method [16], which combines the discriminator-based adversarial training and classifier-based adversarial training to align not only the multiple feature distributions but also the labeled and unlabeled features in the latent space.

- MRAN: The mixup regularized adversarial network [102], which incorporates the domain mixup and category mixup regularizer to enrich the intrinsic feature pattern in the shared latent space to boost the transferability and discriminability of the learned features.

- CAN: The conditional adversarial network [7], which aligns the joint distributions of domain-invariant features and label predictions to improve the discriminability of the learned features. The entropy conditioning mechanism is also incorporated to guarantee the transferability of the learned features.

- CRAL: The co-regularized adversarial learning approach [119], which constructs two diverse training streams, penalizes the disagreements of the outputs of the two streams, and incorporates the virtual adversarial training with entropy minimization on unlabeled data to improve the system performance.

8.4.3 Results

We run the experiments in 5 trial runs and report the average classification accuracy with standard deviation. The experimental results on the Amazon review dataset and FDU-MTL dataset are shown in Table 15 and Table 16, respectively. From Table 15, we observe that the MDAT method can generate competitive performance compared with the state-of-the-art baselines.

For the experimental results on the FDU-MTL dataset, as shown in Table 16, it can be noted that our proposed MDAT method can also yield competitive results compared with the state-of-the-art baselines.

8.5 Further Analysis

8.5.1 Ablation Study

As we adopt the classifier-based adversarial training paradigm in MDAT, we need to validate that the performance gains obtained by our MDAT method mainly come from the margin discrepancy, rather than the classifier-based adversarial training paradigm. We thus investigate one variant: MDAT w ℓ_1. In this variant, we follow [49] to use the ℓ_1 norm to encode the discrepancy induced by the outputs of the two classifiers. The updated adversarial loss $\mathcal{J}'_D$ is formulated as:

$$\mathcal{J}'_D = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x}\sim\mathbb{L}_i\cup\mathbb{U}_i} |\mathcal{C}[\mathcal{F}^i(\mathbf{x})] - \mathcal{C}'[\mathcal{F}^i(\mathbf{x})]| \tag{8.26}$$

where $\mathcal{C}[\cdot]$ and $\mathcal{C}'[\cdot]$ represent the softmax vector generated by $\mathcal{C}$ and $\mathcal{C}'$, respectively.$|\cdot|$ is the ℓ_1 norm function. The variant is also trained by three steps: (1) Calculate $l_C = \mathcal{J}_C$ on the labeled data and update the parameters of $\mathcal{F}_s$, $\mathcal{F}_d^i$, $\mathcal{C}$ and $\mathcal{C}'$ by descending along gradients ∇l_C; (2) Calculate $loss = \mathcal{J}_C + \alpha\mathcal{J}'_D$ on both the labeled and unlabeled data, fix the parameters of $\mathcal{C}$ and $\mathcal{C}'$, and update the parameters of $\mathcal{F}_s$ and $\mathcal{F}_d^i$ by descending along gradients $\nabla loss$; (3) Calculate $l_D = \mathcal{J}'_D$ on both the labeled and unlabeled data, fix the parameters of $\mathcal{F}_s$ and $\mathcal{F}_d^i$, and update the parameters of $\mathcal{C}$ and $\mathcal{C}'$ by ascending along the descents ∇l_D. We conduct the ablation study on the Amazon review dataset and the hyperparameter α is also fixed as $\alpha = 0.5$. The comparison results are shown in Table 17, we can observe that our MDAT can not only outperform the variant on four domains, but also beat it by 1.57% in the average classification accuracy.

Table 17: Ablation Study on the Amazon review dataset

Domain	Books	Dvds	Elec.	Kit.	Avg.
MDAT	84.33	86.07	88.74	90.56	87.43
MDAT w ℓ_1	83.45	85.10	86.30	89.15	85.86

8.5.2 Parameter Sensitivity Analysis

The proposed approach has two hyperparameters α and β. α is used to balance two loss functions, while β controls the margin. We conduct parameter sensitivity analysis on the Amazon review dataset. We first fix $\alpha = 0.5$ and perform experiments with different β values from $\{1, 2, 3, 4, 5\}$. Then we fix $\beta = 4$ and conduct experiments with different α values in the range $\{0.01, 0.1, 0.5, 1.0, 2.0\}$. The experimental results are shown in Figure 13, we can see that α does not influence significantly the performance, we hence set $\alpha = 0.5$ in all our experiments. For β, the average classification accuracy increases quickly with the β value increasing from 1 to 4. The further increase of β will degrade the performance. This suggests that the margin is important in our MDAT method and an appropriate β can make more improvements.

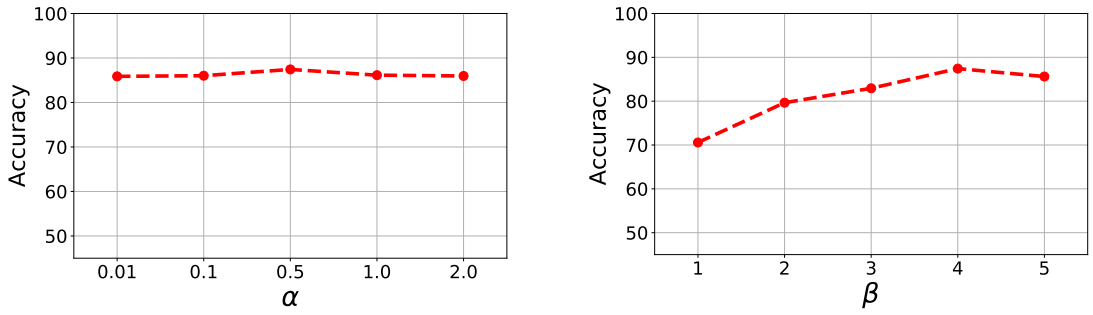


Figure 13: Parameter sensitivity analysis of MDAT

Table 18: Multi-source unsupervised domain adaptation results of MDAT.

Domain	MLP	mSDA	DANN	MDAN	MAN-L2	MAN-NLL	MDAT
Books	76.55	76.98	77.89	78.63	78.45	77.78	**80.76**
DVD	75.88	78.61	78.86	77.97	81.57	82.74	**83.41**
Elec.	84.60	81.98	84.91	85.34	83.37	83.75	**85.27**
Kit.	85.45	84.26	86.39	86.26	85.57	86.41	**86.98**
AVG	80.46	80.46	82.01	82.72	82.24	82.67	**84.11**

8.5.3 Multi-Source Unsupervised Domain Adaptation

When tackling MDTC tasks, the model requires labeled data from each domain. While in real-world applications, many domains may not have any labeled data. Therefore, it is necessary to explore the generalization ability of our method when applied to an unseen domain without labeled data. Standard unsupervised domain adaptation transfers knowledge from one source domain to one target domain, multi-source unsupervised domain adaptation (MS-UDA) allows adaptation from multiple source domains to a single target domain. In the MS-UDA experimental setting, we have multiple source domains with both labeled and unlabeled data and one target domain with unlabeled data. Since the MRAN can leverage unlabeled data across different domains to do representation learning, it is supposed to be effective in the MS-UDA setting. Since the target domain has no labeled data, we only feed the shared features to $\mathcal{C}$, while the domain-specific part is set to 0.

The baselines used in the MS-UDA task include: (1) A MLP trained on the source domains. (2) Two single-source domain adaptation methods, marginalized denoising autoencoder (mSDA) [90] and domain adversarial neural network (DANN) [24]. When

training mSDA and DANN, we combine the data from all source domains into a single source domain. (3) The state-of-the-art MS-UDA methods, multi-source domain adaptation neural networks (MDAN(H) and MADN(S)) [91] and MAN (MAN-L2 and MAN-NLL) [10].

We conduct the MS-UDA experiments on the Amazon review dataset, where three out of four domains are treated as the source domains, and the remaining one is the target one. The classification results of the target domain are reported. The MS-UDA results are shown in Table 18. From Table 18, we can observe that our MDAT method outperforms all baselines on four domains, and beat the second-best approach MAN-NLL method by a margin of 1.44% in terms of the average classification accuracy. The experimental results on the MS-UDA setting demonstrate that the MDAT method has a good ability to generalize to unseen domains.

8.6 Summary

In this chapter, we propose a margin discrepancy-based adversarial training (MDAT) method to tackle the multi-domain text classification (MDTC) problem. This chapter tries to address the sixth MDTC challenge shown in Section 1.2: there is a non-negligible gap between MDTC algorithms and theories. To bridge this gap, we first provide a thorough theoretical analysis for incorporating the margin discrepancy to measure the domain divergence in MDTC and induce a new generalization bound for MDTC based on the margin discrepancy. Then we propose a novel margin discrepancy-based adversarial training (MDAT) method based on the theoretical analysis. To validate the effectiveness of the proposed MDAT approach, We conduct empirical studies on two MDTC benchmarks. The experimental results show that our MDAT method can yield competitive performance compared with the state-of-the-art baselines on both datasets. In the next chapter, we will summarize the contributions

of this thesis and present future research.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This thesis seeks answers to some problems that appear in adversarial learning-based multi-domain text classification methods. The challenges reside in six aspects: (1) How to explicitly model the relationship among the multiple domains; (2) How to further reduce the domain divergence among domains in the latent space; (3) How to use the limited amounts of data from each domain to guarantee the domain-invariance of the shared latent space; (4) When performing adversarial training on the marginal distributions, as the class-aware information is ignored, features belong to different labels can be matched, resulting in weak discriminability of the learned features; (5) Entropy minimization may encourage the model to make over-confident predictions on unlabeled data, deteriorating the discriminability of the learned feature representations; (6) There are no theoretical guarantees to support MDTC algorithms.

In this thesis, we propose five approaches to address these challenges. our contributions are listed as follows:

- We introduce a novel dual adversarial co-learning (DACL) approach which combines domain discrimination based adversarial learning and prediction discrepancy based adversarial learning to enhance the domain-invariance of the learned

feature representations;

- We propose a dual mixup regularized learning (DMRL) method which introduces two forms of mixup-based regularizers to help reduce the distribution discrepancy among multiple domains.

- We propose a conditional adversarial network (CAN) for MDTC which incorporate conditional domain discriminator and entropy conditioning to perform alignment on the joint distributions of shared features and label predictions to improve the system performance.

- We propose a co-regularized adversarial learning (CRAL) method for multi-domain text classification which constructs two diverse shared latent spaces, performs adversarial alignment in each of them, penalizes the disagreements of their predictions on unlabeled data, and incorporates virtual adversarial training with entropy minimization to improve the system performance;

- We propose a margin discrepancy-based adversarial training (MDAT) method which employs the margin discrepancy to measure the domain divergence and guide the adversarial alignment among multiple distributions. Moreover, we provide a strong theoretical analysis to support our MDAT method.

9.2 Future Work

There are multiple potential research aspects to consider for extending the work in the future.

- As matrix can be manipulated to boost the discriminability of the learned features, we can try to investigate how different norms can influence the discriminability of the learned features when they are applied to the batch feature

matrix.

- Currently, there only exist two benchmarks for MDTC, and both benchmarks are used for binary classification. We may try to build a more challenging MDTC benchmark for multi-classification.

- The standard adversarial learning is demonstrated to be unstable and using wasserstein distance as the divergence measurement can effectively make the training procedure of adversarial training more stable. Thus, we may try to introduce wasserstein distance to adversarial multi-domain text classification.

- In the adversarial domain alignment mechanism, domain-invariant features are supposed to be both transferable and discriminative. We can explore at what cost the features obtain the transferability.

List of References

[1] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.

[2] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*, pp. 163–222, Springer, 2012.

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

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[7] Y. Wu, D. Inkpen, and A. El-Roby, "Conditional adversarial networks for multi-domain text classification," in *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pp. 16–27, 2021.

[8] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *ICML*, 2011.

[9] S. J. Pan, J. T. Kwok, Q. Yang, *et al.*, "Transfer learning via dimensionality reduction.," in *AAAI*, vol. 8, pp. 677–682, 2008.

[10] X. Chen and C. Cardie, "Multinomial adversarial networks for multi-domain text classification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1226–1240, 2018.

[11] R. Zheng, J. Chen, and X. Qiu, "Same representation, different attentions: shareable sentence representation learning from multiple tasks," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4616–4622, 2018.

[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[13] S. Li and C. Zong, "Multi-domain sentiment classification," in *Proceedings of ACL-08: HLT, Short Papers*, pp. 257–260, 2008.

[14] Y. Wu, D. Inkpen, and A. El-Roby, "Dual mixup regularized learning for adversarial domain adaptation," in *European Conference on Computer Vision*, pp. 540–555, Springer, 2020.

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

[16] Y. Wu and Y. Guo, "Dual adversarial co-learning for multi-domain text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6438–6445, 2020.

[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv:1310.4546*, 2013.

[18] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 343–351, 2016.

[19] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," *arXiv preprint arXiv:1704.05742*, 2017.

[20] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 1647–1657, 2018.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[22] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

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

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

[26] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[27] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[28] A. Rényi *et al.*, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, The Regents of the University of California, 1961.

[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

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

[31] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

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

[33] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *VLDB*, vol. 4, pp. 180–191, Toronto, Canada, 2004.

[34] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[35] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*, pp. 153–171, Springer, 2017.

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

[37] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," *arXiv preprint arXiv:1702.08811*, 2017.

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

[39] M. Long and J. Wang, "Learning multiple tasks with deep relationship networks," *arXiv preprint arXiv:1506.02117*, vol. 2, no. 1, 2015.

[40] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*, pp. 2208–2217, PMLR, 2017.

[41] Y. Zhang, N. Wang, S. Cai, and L. Song, "Unsupervised domain adaptation by mapped correlation alignment," *IEEE Access*, vol. 6, pp. 44698–44706, 2018.

[42] Y. Wang, W. Li, D. Dai, and L. Van Gool, "Deep domain adaptation by geodesic distance minimization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2651–2657, 2017.

[43] Z. Zhang, M. Wang, Y. Huang, and A. Nehorai, "Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3437–3445, 2018.

[44] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

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

[46] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

[47] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, 2016.

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

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

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

[51] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[52] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8503–8512, 2018.

[53] H. Zou, Y. Zhou, J. Yang, H. Liu, H. P. Das, and C. J. Spanos, "Consensus adversarial domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 5997–6004, 2019.

[54] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[55] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2239–2247, 2019.

[56] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *International conference on machine learning*, pp. 1081–1090, PMLR, 2019.

[57] M. Li, Y.-M. Zhai, Y.-W. Luo, P.-F. Ge, and C.-X. Ren, "Enhanced transport distance for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13936–13944, 2020.

[58] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

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

[60] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[61] L. B. Smith, "Learning to recognize objects," *Psychological Science*, vol. 14, no. 3, pp. 244–250, 2003.

[62] S. Parameswaran and K. Q. Weinberger, "Large margin multi-task metric learning," *Advances in neural information processing systems*, vol. 23, 2010.

[63] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Multi-task learning via conic programming," *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[64] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[65] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors.," *Journal of Machine Learning Research*, vol. 8, no. 1, 2007.

[66] H. Daumé III, "Bayesian multitask learning with latent hierarchies," *arXiv preprint arXiv:0907.0783*, 2009.

[67] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 733–742, 2010.

[68] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *ICML*, 2011.

[69] A. Kumar and H. Daume III, "Learning task grouping and overlap in multi-task learning," *arXiv preprint arXiv:1206.6417*, 2012.

[70] K. Crammer and Y. Mansour, "Learning multiple tasks using shared hypotheses," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1475–1483, 2012.

[71] W. Lu, H. L. Chieu, and J. Löfgren, "A general regularization framework for domain adaptation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 950–954, 2016.

[72] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003, 2016.

[73] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1930–1939, 2018.

[74] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4106–4115, 2019.

[75] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2873–2879, 2016.

[76] Q. Yang and L. Shang, "Multi-task learning with bidirectional language models for text classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[77] X. Su, R. Li, and X. Li, "Multi-domain transfer learning for text classification," in *CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 457–469, Springer, 2020.

[78] F. Wu and Y. Huang, "Sentiment domain adaptation with multiple sources," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 301–310, 2016.

[79] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447, 2007.

[80] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, 2014.

[81] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[82] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[83] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.

[84] L. Duan, D. Xu, and I. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 504–518, 2012.

[85] F. Wu and Y. Huang, "Collaborative multi-domain sentiment classification," in *2015 IEEE International Conference on Data Mining*, pp. 459–468, IEEE, 2015.

[86] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[87] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[88] J. Zhou, J. Chen, and J. Ye, "MALSAR: Multi-task learning via structural regularization," *Arizona State University*, vol. 21, 2011.

[89] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 912–921, 2015.

[90] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1627–1634, 2012.

[91] H. Zhao, S. Zhang, G. Wu, J. Costeira, J. Moura, and G. Gordon, "Multiple source domain adaptation with adversarial training of neural networks," *arXiv preprint arXiv:1705.09684*, 2017.

[92] A. Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. Feris, B. Freeman, and G. Wornell, "Co-regularized alignment for unsupervised domain adaptation," in *Advances in Neural Information Processing Systems*, pp. 9345–9356, 2018.

[93] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[94] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," *Advances in neural information processing systems*, vol. 13, 2000.

[95] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Machine Learning*, pp. 6438–6447, 2019.

[96] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[97] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[98] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8050–8058, 2019.

[99] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in neural information processing systems*, pp. 529–536, 2005.

[100] J. A. Aslam and V. Pavlu, "Query hardness estimation using jensen-shannon divergence among multiple scoring functions," in *European conference on information retrieval*, pp. 198–209, Springer, 2007.

[101] L. Brillouin, *Science and information theory*. Courier Corporation, 2013.

[102] Y. Wu, D. Inkpen, and A. El-Roby, "Mixup regularized adversarial networks for multi-domain text classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7733–7737, IEEE, 2021.

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

[104] B. Smith and G. Linden, "Two decades of recommender systems at amazon. com," *Ieee internet computing*, vol. 21, no. 3, pp. 12–18, 2017.

[105] E. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision support systems*, vol. 50, no. 3, pp. 559–569, 2011.

[106] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[107] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[108] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation.," in *AISTATS*, vol. 2005, pp. 57–64, Citeseer, 2005.

[109] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," *arXiv preprint arXiv:1802.08735*, 2018.

[110] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," in *International Conference on Learning Representations*, no. 6, 2018.

[111] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," in *International Joint Conference on Artificial Intelligence*, pp. 3635–3641, 2019.

[112] T. Van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[113] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *International Conference on Machine Learning*, pp. 7404–7413, PMLR, 2019.

[114] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[115] Y. Lin, "A note on margin-based loss functions in classification," *Statistics & probability letters*, vol. 68, no. 1, pp. 73–82, 2004.

[116] V. Koltchinskii, D. Panchenko, *et al.*, "Empirical margin distributions and bounding the generalization error of combined classifiers," *The Annals of Statistics*, vol. 30, no. 1, pp. 1–50, 2002.

[117] A. Galbis and M. Maestre, *Vector analysis versus vector calculus*. Springer Science & Business Media, 2012.

[118] M. Talagrand, *Upper and lower bounds for stochastic processes: modern methods and classical problems*, vol. 60. Springer Science & Business Media, 2014.

[119] Y. Wu, D. Inkpen, and A. El-Roby, "Co-regularized adversarial learning for multi-domain text classification," *arXiv preprint arXiv:2201.12796*, 2022.