LEARNING ADAPTIVE DEEP REPRESENTATIONS FOR FEW-TO-MEDIUM SHOT IMAGE CLASSIFICATION

by

Xiang Jiang

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
January 2021

Dedicated to my beloved parents and grandparents,
for their unconditional love.

Table of Contents

List of Tables vi

List of Figures vii

Abstract ix

List of Abbreviations Used x

Acknowledgements xi

Chapter 1 Introduction 1

1.1 Data Scarcity and Distribution Mismatch in Deep Learning 3
1.1.1 Data Scarcity 3
1.1.2 Distribution Mismatch 5

1.2 Contributions 8

1.3 Thesis Organization 11

Chapter 2 Background and Related Work 13

2.1 Domain Adaptation 13
2.1.1 A Brief Theory of Domain Adaptation 13
2.1.2 Adversarial Learning for Domain Adaptation 15
2.1.3 Related Work for Domain Adaptation 16

2.2 Meta-learning 18
2.2.1 Meta-learning Problem Formulation 19
2.2.2 Related Work for Deep Learning- Based Meta-learning 21

Chapter 3 Implicit Class-Conditioned Domain Alignment for Unsupervised Domain Adaptation 24

3.1 Introduction 24

3.2 Method 25
3.2.1 Theoretical Motivations 25
3.2.2 Implicit Class-Conditioned Domain Alignment 28
3.2.3 Class-Aligned Sampling Strategy 29
3.2.4 Integrating Implicit Alignment into Classifier-Based Domain Discrepancy Measure 31

3.3 Experiments on Standard Domain Adaptation Datasets 32
3.3.1 Setup 32
3.3.2 Results and Discussions 33

3.4 Synthetic Experiments on MNIST and SVHN 40
3.4.1 Setup 41
3.4.2 Results on Matched Class Distribution 41
3.4.3 Results on Mismatched Class Distribution 42

3.5 Supplementary Empirical Results 43
3.5.1 Additional Evaluation Measures on Office-Home 43
3.5.2 Ablation on Alignment Options 44
3.5.3 Learning Curve 44
3.5.4 Computational Efficiency 45
3.5.5 The Impact of Batch Size 46
3.5.6 Empirical Class Diversity 47
3.5.7 VisDA2017 Dataset 48

3.6 Limitations 48

3.7 Conclusion 49

Chapter 4 Learning to Learn with Conditional Class Dependencies 50

4.1 Introduction 50

4.2 Method 51
4.2.1 Metric Space as Conditional Information 53
4.2.2 Conditionally Transformed Convolutional Block 53
4.2.3 Multitask learning of CBN 55
4.2.4 Training Details 56

4.3 Experiments 56
4.3.1 Results 56
4.3.2 The effect of conditional transformation 58

4.4 Limitations 59

4.5 Conclusions 60

Chapter 5 Task Adaptive Metric Space
for Medium-Shot Medical Image Classification 61

5.1 Introduction 61

5.2 Background and Motivation 62

5.3 Task Adaptive Metric Space
for Improved Medium-shot Generalization 64

5.4 Empirical Results 66
5.4.1 Data and Evaluation Setup 66
5.4.2 Results and Discussions 67

5.5 Limitations 69

5.6 Conclusions 70

Chapter 6 Conclusion and Future Research 71

6.1 Conclusion 71

6.2 Future Research 72
6.2.1 Meta-learning Domain Adaptation 72
6.2.2 Privacy-Preserving Domain Adaptation 74

6.3 Grounded meta-learning 74

Bibliography 76

List of Tables

3.1 Per-class average accuracy on Office-Home dataset with RS-UT label shifts (ResNet-50). 34

3.2 Accuracy with 95% confidence interval (%) on Office-31 (standard) for unsupervised domain adaptation (ResNet-50). 36

3.3 Accuracy (%) on Office-Home (standard) for unsupervised domain adaptation (ResNet-50). 36

3.4 The impact of different implicit alignment options 40

3.5 Per-class average accuracy (%) with *matched prior* where $p_S(y) = p_T(y)$. 42

3.6 Per-class average accuracy (%) with *mismatched prior* where $p_S(y) \neq p_T(y)$ and both domains are imbalanced. 42

3.7 Per-class average accuracy (%) with *mismatched prior* where $p_S(y) \neq p_T(y)$: source-balance, target-imbalanced. 43

3.8 Per-class average accuracy (%) with *mismatched prior* where $p_S(y) \neq p_T(y)$: source-imbalance, target-balanced. 43

3.9 Additional evaluation measures on Office-Home (%) with ResNet-50. 43

3.10 The impact of different implicit alignment options. 44

3.11 The impact of pseudo-label update frequency on Ar→Cl, Office-Home (standard). 45

3.12 The impact of batch size on target domain accuracy (%) 46

3.13 Target domain accuracy (%) on VisDA2017 (synthetic→real) . . 48

4.1 *mini*ImageNet classification accuracy with 95% confidence intervals. 57

5.1 OCT test accuracy with “conv4” architecture (%) 67

List of Figures

1.1 Definition of domain shift. 6

3.1 Illustration of the domain discriminator shortcut. 28

3.2 The proposed implicit class-conditioned domain alignment framework. 29

3.3 (a) Source and target class distribution of Office-Home (RS-UT). (b) Accuracy comparison between Office-Home (RS-UT) and Office-Home (balanced) for Rw→Pr. 34

3.4 The impact of class diversity and alignment. 37

3.5 The impact of pseudo-label errors on implicit and explicit alignment, Office-Home (standard). 39

3.6 Interactions between within-domain class imbalance and between-domain class distribution shift. 41

3.7 Learning curve of the target domain accuracy for Pr→Rw, Office-Home (RS-UT). 45

3.8 Empirical class diversity while training A→W (Office-31) with batch size 31. 47

4.1 Overview of Conditional class-Aware Meta-Learning. 52

4.2 Conditionally transformed convolutional block. 54

4.3 CBN shared architecture 55

4.4 t-SNE visualization of the learned metric space colored by category. 57

4.5 PCA visualization of feature maps from the last convolutional layer colored by category. 59

5.1 t-SNE visualization of sampled test data before adaptation . . . 67

5.2 t-SNE visualization of sampled test data after adaptation . . . 67

5.3 Metric adaptation steps. 69

5.4 Impact of model architecture. 69

6.1 Meta-learning domain adaptation 72

Abstract

In real-world applications, the environment in which a machine learning system is deployed tends to change due to many factors, such as sample selection bias, prior probability mismatch, and domain shift. This makes it difficult to reliably generalize deep learning models from the training set to real-world scenarios. In addition, data scarcity frequently arises from a large number of applications where annotating data is expensive or requires specialized expertise. As machine learning applications progress into more complex tasks that require models with magnitudes higher Vapnik–Chervonenkis dimensions, more labeled training data are necessary to maintain the same upper bound for the test error. To this end, there is an ever-increasing need for sample efficient learning systems that can adapt to changing environments. This thesis aims to study the generalization of deep learning models in the presence of distribution mismatch and data scarcity.

We first study unsupervised domain adaptation, an emerging field of semi-supervised learning that aims to address domain shift with labeled data in the source domain and unlabeled data in the target domain. We propose implicit class-conditioned domain alignment to address between-domain class distribution shift. A theoretical analysis is provided to justify the proposed method by decomposing the empirical domain divergence into class-aligned and class-misaligned divergence, and we show that class-misaligned divergence is detrimental to domain adaptation. We show that our method offers consistent improvements for different adversarial adaptation algorithms.

We also propose two meta-learning methods to bridge the gap between gradient and metric-based methods. The first proposal is Conditional class-Aware Meta-Learning where we introduce a metric space to modulate the image representation of a model, resulting in better separated feature representations. Motivated by the discrepancy of the number of training examples between few-shot and real-world medical datasets, the second proposal is to extend few-shot learning to few-to-medium-shot learning. The proposed Task Adaptive Metric Space uses gradient-based fine-tuning to adjust parameters of the metric space to provide more flexibility to metric-based methods. The method adjusts the metric space to better reflect examples of a new medical classification task.

List of Abbreviations Used

DANN	domain adversarial neural networks
GAN	Generative adversarial networks
MAML	Model-Agnostic Meta-Learning
MDD	margin disparity discrepancy
MMD	Maximum Mean Discrepancy
MR	magnetic resonance
SVM	Support Vector Machine
t-SNE	t-distributed stochastic neighbor embedding
UDA	Unsupervised Domain Adaptation
VC-dimension	Vapnik–Chervonenkis dimensions

Acknowledgements

First and foremost, I am incredibly grateful to my supervisors, Dr. Stan Matwin and Dr. Daniel Silver, for their invaluable support during my Ph.D. study. Dr. Silver introduced me to the beautiful field of machine learning in 2013. His philosophical thoughts about machine learning built a fundamental compass that helps me navigate the machine learning world. Dr. Matwin is a Canada Research Chair, a generous and strategic thinker who helped and encouraged me to participate in research training in the industry that I am forever grateful. I also appreciate the support from my respected Ph.D. committee members: Dr. Mark Schmidt, Dr. Sageev Oore, Dr. Luis Torgo, and Dr. Thomas Trappenberg.

I am deeply thankful to Dr. Mohammad Havaei for his immense knowledge and generous mentorship—my great pleasure to have the opportunity to work with him. At Imagia, I am fortunate to work with and learn from Thomas Vincent, Andrew Jesson, Dr. Lisa Di Jorio, Dr. Nicolas Chapados, Dr. Qicheng Lao, Dr. Gabriel Chartrand, Tanya Nair, Dr. Hassan Chouaib, Dr. Cecile Low-Kam, Francis Dutil, Dr. Kim Phan, Dr. Nicolas Guizard, Dr. Sina Hamidi, Tess Berthier, Philippe Lacaille, Dr. Martine Bertrand and more. I have learned unique bits of knowledge from each of you!

I would like to express my sincere appreciation to all of my colleagues at the Institute for Big Data Analytics. The institute is a warm family from all over the world and thank you for a cherished time spent together in the lab—Dr. Ahmad Pesaranghader, Dr. Behrouz Haji Soleimani, Habibeh Naderi, Dr. Fateha Khanam Bappee, Dr. Jianzhe Zhao, Dr. Amílcar Soares Júnior, Lucas May Petry, Dr. Oliver S. Kirsebom, Fábio Frazão, Bo Liu, Farshid Varno, Pedram Adibi, Mark Thomas, Xuhui Liu, Dr. Xuan Liu, Dr. Xiaoguang Wang, Jie Mei, Dr. Aaron Gerow, Lulu Huang, Baifan Hu, Hossein Sarshar, David Samuel and more.

Last but not the least, I would like to thank my parents and grandparents, who set me off on the road, for their unconditional love, encouragement, and support in my life.

Chapter 1

Introduction

Machine learning is a sub-field of artificial intelligence that aims to teach computers to perform specific tasks without programming the knowledge into the computer explicitly. More formally, Mitchell et al. [1997] defined a well-posed machine learning problem where "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." Machine learning typically comprises of two steps: (i) *learning*: a model learns from the training data in a maximum likelihood [Aldrich et al., 1997] or maximum a posteriori [Murphy, 2012] manner where the learned parameters represent patterns uncovered from the training data, and (ii) *inference*: the learned model is deployed on unseen data to infer statements of interest such as how to make decisions for the new data.

In a traditional machine learning setup, there exists a task $\boldsymbol{\mathcal{D}}$, such as classifying handwritten digits into different categories, identifying outliers from data or playing the game Go. The examples of this task are typically divided into a training set $\boldsymbol{\mathcal{D}}^{\text{train}}$, a validation set $\boldsymbol{\mathcal{D}}^{\text{valid}}$ and a test set $\boldsymbol{\mathcal{D}}^{\text{test}}$. A model is trained on $\boldsymbol{\mathcal{D}}^{\text{train}}$ to optimize for parameters θ that minimize some loss $\mathcal{L}$,

$$\theta^* = \text{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\mathcal{D}}^{\text{train}};\theta). \tag{1.1}$$

The validation set $\boldsymbol{\mathcal{D}}^{\text{valid}}$ is used for hyperparameter selection and the generalization error of the model is estimated on the test set $\boldsymbol{\mathcal{D}}^{\text{test}}$.

Machine learning has evolved into an exciting field where the learning task $\boldsymbol{\mathcal{D}}$ can take many different forms that are tailored to different types of problems.

1. In supervised learning, the model is trained on labeled data $\boldsymbol{\mathcal{D}}^{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ where each example x_i is associated with a target label y_i. Supervised learning is the most widely used form of machine learning with applications to image classification

and machine translation. Representative methods include Support Vector Machine (SVM) [Cortes and Vapnik, 1995], random forest [Ho, 1995], gradient boosting [Breiman, 1997; Friedman, 2001], and neural networks [Goodfellow et al., 2016].

2. In unsupervised learning, the model only has access to unlabeled data $\boldsymbol{\mathcal{D}} = \{(x_i)\}_{i=1}^{N}$ and the goal is to uncover patterns of the data to further the understanding. Clustering and image reconstruction are typical examples of unsupervised learning. Representative methods include K-means [MacQueen et al., 1967], t-distributed stochastic neighbor embedding (t-SNE) [Maaten and Hinton, 2008], autoencoders [Kingma and Welling, 2013; Vincent et al., 2010] and generative adversarial networks [Goodfellow et al., 2014].

3. In semi-supervised learning, the model has access to labeled data together with some unlabeled data. Assumptions about the relationship between labeled and unlabeled data, or assumptions on the underlying distribution of the data are necessary to make use of the unlabeled data [Chapelle et al., 2009]. Representative methods include self-training [Fralick, 1967; Scudder, 1965], transductive inference [Vapnik, 2006] and expectation-maximization [Dempster et al., 1977].

4. In reinforcement learning, the model learns the best sequence of actions to maximize the expected cumulative reward in the future [Kaelbling et al., 1996]. Different from supervised learning where the model learns from labeled data directly, reinforcement learning explores the world through a sequence of actions and uses the reward signal from the environment at the end of the action sequence to learn to maximize the award. In March 2016, an reinforcement learning program AlphaGo beat a 9-dan professional in the game GO using deep reinforcement learning with monte-carlo tree search [Silver et al., 2016].

The objective of this thesis is on the generalization of deep learning models in the presence of data scarcity and distribution mismatch. The scope of this thesis is to study supervised learning and semi-supervised learning approaches of deep learning based image classification tasks where a computer program is tasked to not only *learn* from experience but also *adapt* to environments. This is motivated by the fact that a human can learn and adapt to new concepts efficiently from a small number of examples by making use of their past learning experience.

1.1 Data Scarcity and Distribution Mismatch in Deep Learning

Learning from data is an inductive process where the learning system aims to derive general principles from the observed data. However, inductive reasoning is susceptible to data scarcity and distribution mismatch, which are crucial to the reliability of the evidential inference.

1.1.1 Data Scarcity

Data scarcity frequently arises from a large number of real-world applications where labeling data is expensive or requires specialized training and expertise, such as annotating medical images. Besides, as the machine learning research community progresses into more complex tasks that require models with higher Vapnik–Chervonenkis dimensions (VC-dimension) [Blumer et al., 1989], more labeled training data are necessary to maintain the same upper bound for the test error. As a result, there is an ever-increasing need for sample efficient learning systems. This is especially the case for deep learning. Although deep learning models have proven to be highly effective when trained on vast amounts of data, they fail to generalize from very few examples.

Transfer learning has become one of the most popular approach when faced with data scarcity [Pan and Yang, 2009; Silver et al., 2008; Zhuang et al., 2020]. Transfer learning usually involves a set of tasks that can facilitate learning in a unidirectional or bidirectional manner. The ubiquitous transfer learning approach in deep learning is parameter transfer, where we train a model on one task from large amounts of data and use the learned parameters to initialize another task with fewer data. Open-source deep learning frameworks have accelerated the adoption of transfer learning by providing pre-trained models on largescale datasets in the form of model checkpoints, thereby removing the need to pre-train models on computationally expensive datasets. ImageNet [Russakovsky et al., 2015] and BERT [Devlin et al., 2018] are prominent examples of the success of transfer learning, where a pre-trained image classification model or language model can be transferred to a wide range of downstream tasks, such as detection, segmentation, and natural language inference. In contrast to conventional transfer learning methods that aim to fine-tune a pre-trained model, meta-learning* systems are trained by being exposed to a large number of tasks and evaluated in their ability to learn new tasks effectively. In meta-training, learning happens at two levels: a meta-learner that learns across many tasks and a base-learner that optimizes for each individual task.

In recent years, meta-learning [Bengio et al., 1992; Branco et al., 2018; Mitchell and Thrun, 1993; Schmidhuber, 1987; Vilalta and Drissi, 2002] has evolved into a promising research direction to address data scarcity. Instead of learning individual tasks independently, meta-learning integrates the learning of many tasks in a hierarchical manner so as to acquire meta-knowledge across many tasks. This is akin to a Gaussian process [Kac and Siegert, 1947] that learns a distribution of functional predictors but only a finite set of functions. In contrast to a standard supervised learning setup where the goal is to learn a single model from a task $\mathcal{D}$, a meta-learner aims to learn from a set of tasks $\{\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_S\}$ to get better at learning new tasks. At the meta-level, each task $\mathcal{D}_i$ is treated as a training example.

Meta-learning has become an important approach for few-shot learning. Previous work on deep learning based meta-learning can be summarized as: learning representations that encourage fast adaptation on new tasks [Finn et al., 2017a,b], learning universal learning procedure approximators [Hochreiter et al., 2001; Mishra et al., 2017; Santoro et al., 2016; Vinyals et al., 2016], learning to generate model parameters conditioned on training examples [Gomez and Schmidhuber, 2005; Ha et al., 2016; Munkhdalai and Yu, 2017], learning optimization algorithms [Andrychowicz et al., 2016; Bengio et al., 1992; Li and Malik, 2017; Ravi and Larochelle, 2016], and learning a metric space for distance-based inference [Oreshkin et al., 2018; Ren et al., 2018; Snell et al., 2017b].

In this thesis, two meta-learning approaches are proposed to bridge the gap between gradient-based methods and metric-based methods. The first proposal is Conditional class-Aware Meta-Learning (CAML) motivated by a core challenge in gradient-based meta-learning, wherein the quality of gradient information is key to fast generalization: it is known that gradient-based optimization fails to converge adequately when trained

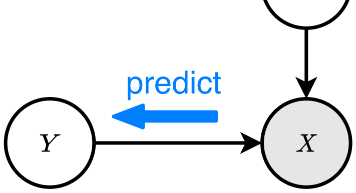
*Meta-learning [Bengio et al., 1992; Branco et al., 2018; Ling and Sheng, 2010; Mitchell and Thrun, 1993; Schmidhuber, 1987; Vilalta and Drissi, 2002] has been studied extensively in the machine learning literature. Although "different researchers hold different views of what the term meta-learning exactly means", the common goal of meta-learning is to "exploit the knowledge of learning (meta-knowledge) to improve the performance of learning algorithms" [Vilalta and Drissi, 2002]. In this thesis, the term "meta-learning" is restricted to deep learning based approaches for learning across different tasks.

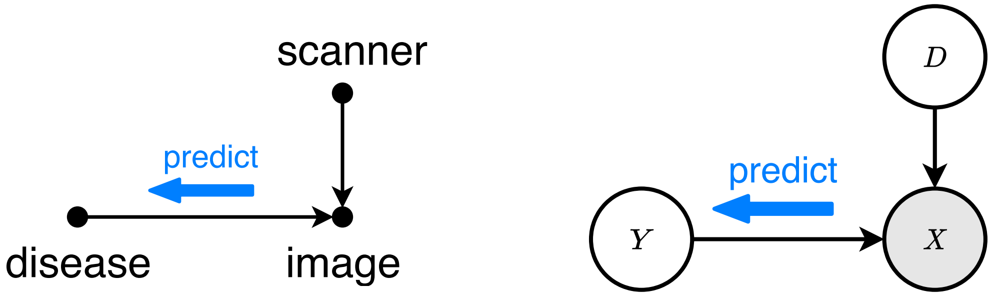
from only a few examples [Ravi and Larochelle, 2016], hampering the effectiveness of gradient-based meta-learning techniques. We hypothesize that under such circumstances, introducing a metric space trained to encode regularities of the label structure can impose global class dependencies on the model. This class structure can then provide a high-level view of the input examples, in turn leading to learning more disentangled representations.

The second proposal is to extend few-shot learning to few-to-medium-shot learning for more realistic evaluations of real-world datasets. This is motivated by the discrepancy of the number of training examples between few-shot tasks and real-world medical datasets. While in the literature meta-learning mostly deals with tasks with only a few training examples, i.e., five or fewer, datasets in the medical domain tend to have tens to a few hundred labeled examples. With this in mind, we propose to evaluate representative meta-learning methods under different amounts of data per class, so as to better understand their generalization properties. We choose key advances in meta-learning—*gradient-based* [Finn et al., 2017a] and *metric-based* [Snell et al., 2017b] methods—to establish the baseline performances. We empirically evaluate and analyze the baseline meta-learning methods through the lens of bias-variance tradeoff. Our analysis suggests gradient-based methods tend to overfit few-shot datasets while metric-based methods tend to underfit medium-shot datasets. To get the best of both worlds for the bias-variance equilibrium, we propose Task Adaptive Metric Space (TAMS) that uses gradient-based fine-tuning to adjust parameters of the metric space so that distances between examples in the medical dataset can better reflect their semantics.

1.1.2 Distribution Mismatch

In real-world applications, the environment in which a machine learning system is deployed tends to change due to a large number of factors. As an example, magnetic resonance () scanners are built with different strength of the magnet such as 1.5 Tesla and 3 Tesla. The strength of the magnet has a direct impact on the appearance of MR images: the higher the strength, the more signals can be captured from the body which might also create artifacts in the image. If we trained an accurate machine learning model from a large number of annotated MRI scans on a 3-Tesla machine, the same model will not be directly applicable to images acquired with a 1.5-Tesla machine due to the distribution mismatch of the images acquired from different machines. To

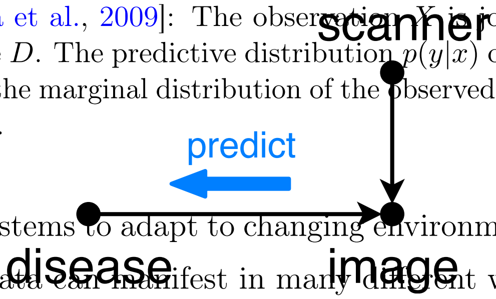




(a): Example of domain shift.

(b): Domain shift.

Figure 1.1: Domain shift [Quionero-Candela et al., 2009]: The observation X is jointly determined by the label Y and domain variable D. The predictive distribution $p(y|x)$ of the labels given the observations is unchanged, but the marginal distribution of the observed data $p(x)$ varies as we change the domain variable D.



this end, it is desirable to enable learning systems to adapt to changing environments. The mismatch between training and test data can manifest in many different ways, such as sample selection bias [Heckman, 1979; Torralba et al., 2011], class distribution mismatch (class imbalance) [Chawla, 2009; Japkowicz and Stephen, 2002; Webb and Ting, 2005], and covariate shift [Shimodaira, 2000].

Unsupervised Domain Adaptation (UDA) is an emerging field of semi-supervised learning that aims to address a specific type of distribution mismatch named "domain shift" which is intuitively defined as "changes in the measurement system".

In the aforementioned example shown in Figure 1.1 (a), observations X from different MR scanners are jointly determined by the disease label Y together with the scanner label D. We assume the causal mechanism from disease labels to scans are the same, but the marginal distribution of the scans are different on different scanners due to domain shift. The goal of domain adaptation is to uncover the predictive function to predict disease from images that is independent of the measurement system. More formally, as shown in Figure 1.1 (b), we use X to denote the observed variable. The observation X is jointly determined by the class label Y and the domain variable D. We assume both domains share the same labeling function $f_S = f_T$ that is independent of the domain variable D, where $Y = f(X)$. In other words, the predictive distribution of the labels given the observations is unchanged $p(y|x)$ under domain shift, but the marginal distribution of the observed data $p(x)$ varies as we change values of the domain variable D, i.e., a change of measurement system.

More concretely, in unsupervised domain adaptation, the model is trained on labeled samples $\{(x_i, f_S(x_i))\}_{i=1}^{n}$ from the source domain where $x_i \sim \mathcal{D}_S$, together with unlabeled samples $\{x_j\}_{j=1}^{m}$ from the target distribution where $x_j \sim \mathcal{D}_T$. The goal is to obtain a model $h \in \mathcal{H}$ which learns domain-invariant representations while simultaneously minimizing the classification error on $\mathcal{D}_S$.

Considerable progress has been made in domain adaptation since 2006. Early approaches include Kernel Maximum Mean Discrepancy (MMD) [Borgwardt et al., 2006] which uses a kernel-based approach to measure the distribution mismatch based on the mean discrepancy of two distributions. MMD has been extended to deep learning by means of a loss function that aims to minimize the distribution mismatch between the source and target domains at the feature level [Tzeng et al., 2014]. The main limitation for MMD-based approaches is that they only match a single mode, i.e., the mean, of source and target distributions and is ineffective in dealing with data with multimodal distributions. Another early approach to domain adaptation is sample re-weighting such that reweighted source and target data are close in reproducing kernel Hilbert space [Huang et al., 2007; Jiang and Zhai, 2007]. The main challenge is that it is difficult to estimate the density ratio of distributions in high dimensional input space, such as images, to provide a proper re-weighting scheme.

Generative adversarial networks (GAN) [Goodfellow et al., 2014] revolutionized unsupervised domain adaptation. The minimax method has emerged as the prevalent approach with the goal to learn domain-invariant representations such that the domain discriminator can not distinguish whether the marginal feature distribution is from the source or the target domain [Ganin et al., 2016]. While substantial progress has been made in adversarial approaches for domain adaptation, they tend to focus on marginal distribution alignment in the feature space, and less emphasis is made on discovering the label distributions of the source and target domains. However, in real-world applications, it is very common to have class imbalance within each domain. It is also common to have class distribution shift between different domains where the marginal distribution of classes varies between domains. This necessitates the incorporation of label space distribution into domain adaptation models.

In this thesis, we propose *Implicit* Class-Conditioned Domain Alignment that removes the assumptions of identical distributions in the label space. The proposed approach uses model's predictions on the target domain as pseudo-labels *implicitly* to *sample* class-conditioned data in a way that maximally aligns the joint distribution between features and labels. The primary advantage of the sampling-based implicit domain alignment is the ability to address between-domain class distribution shift.

1.2 Contributions

The outcome of this research is a set of deep learning methods for dealing with data scarcity and distribution mismatch. The contributions of this thesis are [†]:

1. We propose implicit class-conditioned domain alignment to address between-domain class distribution shift, which also overcomes the limitations of explicit domain alignment. We show that our method offers consistent improvements for different adversarial adaptation algorithms: both DANN and MDD.

2. We provide a theoretical analysis by decomposing the empirical domain divergence into class-aligned and class-misaligned divergence. We show that class-misaligned divergence is detrimental to domain adaptation. We identify a domain discriminator shortcut function that interferes with adversarial domain adaptation. The shortcut could bypass the optimization for domain-invariant representations, but rather optimize for a shortcut function that is independent of the covariate contributing to the domain difference.

3. We design extensive experiments to further demonstrate the effectiveness of the proposed method under different challenges. The class distributions of SVHN and MNIST are synthetically manipulated to simulate various interactions between within-domain class imbalance and between-domain class distribution shift. We report state-of-the-art UDA performance under extreme within-domain class imbalance and between-domain class distribution shift, as well as competitive results on standard UDA tasks compared with state-of-the-art adversarial domain adaptation approaches.

4. We propose a meta-learning framework that makes use of structured class information in the form of a metric space to modulate representations in few-shot

[†] Parts of this thesis is published earlier in our research articles [Jiang et al., 2017, 2018, 2019, 2020].

learning tasks. Model-Agnostic Meta-Learning (MAML) gives us a way to do model-agnostic initialization of weights and prototypical networks gives a good way to take into account similarity among classes. We show experimentally that our proposed algorithm learns how to disentangle, or separate, representation between different classes and achieves competitive results on the *mini*ImageNet benchmark.

5. We propose *medium-shot learning* that aligns meta-learning with realistic situations of medical image classification. We establish baseline evaluation procedures for meta-learners in various situations to better understand their generalization properties.

6. Through bias-variance analysis, we propose a new meta-learning method—Task Adaptive Metric Space—that takes advantage of both gradient-based and metric-based methods. We show that TAMS outperforms the meta-learning baselines.

Below is a list of thesis-related papers published during my graduate work:

1. Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. "Implicit Class-Conditioned Domain Alignment for Unsupervised Domain Adaptation." In International Conference on Machine Learning (ICML), 2020.

2. Qicheng Lao, Xiang Jiang, Mohammad Havaei, and Yoshua Bengio. "Continuous Domain Adaptation with Variational Domain-Agnostic Feature Replay." IEEE Transactions on Neural Networks and Learning Systems (TNNLS), 2021.

3. Xiang Jiang, Mohammad Havaei, Farshid Varno, Gabriel Chartrand, Nicolas Chapados, and Stan Matwin. "Learning to learn with conditional class dependencies." In International Conference on Learning Representations (ICLR), 2018.

4. Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. "Attentive Task-Agnostic Meta-Learning for Few-Shot Text Classification." NeurIPS Workshop on Meta-Learning, 2018.

5. Xiang Jiang, Liqiang Ding, Mohammad Havaei, Andrew Jesson, and Stan Matwin. "Task Adaptive Metric Space for Medium-Shot Medical Image Classification." In

International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), pp. 147-155. Springer, 2019.

Below is a list of papers published during my graduate work that are not directly related to this thesis:

1. Xiang Jiang, Erico N. de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L. Silver, and Stan Matwin. “Trajectorynet: An embedded gps trajectory representation for point-based classification using recurrent neural networks.” In Annual International Conference on Computer Science and Software Engineering (CASCON), pp. 192-200. IBM Corp., 2017 (Best Paper Award).

2. Xiang Jiang, Xuan Liu, Erico N. de Souza, Baifan Hu, Daniel L. Silver, and Stan Matwin. “Improving point-based AIS trajectory classification with partition-wise gated recurrent units.” In International Joint Conference on Neural Networks (IJCNN), pp. 4044-4051. IEEE, 2017.

3. Xiang Jiang, Erico N de Souza, Xuan Liu, Behrouz Haji Soleimani, Xiaoguang Wang, Daniel L Silver, Stan Matwin, “Partition-wise Recurrent Neural Networks for Point-based AIS Trajectory Classification”, in European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2017.

4. Xiang Jiang, Daniel L Silver, Baifan Hu, Erico N de Souza, Stan Matwin, “Fishing activity detection from AIS data using autoencoders”, in Canadian Conference on Artificial Intelligence, 2016.

5. Baifan Hu, Xiang Jiang, Erico N de Souza, Ronald Pelot, Stan Matwin, “Identifying fishing activities from AIS data with conditional random fields”, in Federated Conference on Computer Science and Information Systems (FedCSIS), 2016.

6. Mei Jie, Xiang Jiang, Aminul Islam, Abidalrahman Moh’d, and Evangelos Milios. “Integrating Global Attention for Pairwise Text Comparison.” In Proceedings of the ACM Symposium on Document Engineering, 2018.

1.3 Thesis Organization

The overall structure and organization of the thesis is as follows.

Chapter 2 provides the background and related work for domain adaptation and meta-learning.

Chapter 3 introduces an approach for unsupervised domain adaptation—with a strong focus on practical considerations of between-domain class distribution shift—from a class-conditioned domain alignment perspective. We show theoretically that the proposed implicit alignment provides a more reliable measure of empirical domain divergence which facilitates adversarial domain-invariant representation learning, that would otherwise be hampered by the class-misaligned domain divergence. We show that our proposed approach leads to superior UDA performance under extreme within-domain class imbalance and between-domain class distribution shift, as well as competitive results on standard UDA tasks. We further demonstrate that implicit alignment overcomes the critical limitations of pseudo-label bias by removing the need for explicit optimization of model parameters from pseudo-labels. We emphasize that the proposed method is robust to pseudo-label bias, simple to implement, has a unified training objective, and does not require additional parameter tuning.

Chapter 4 proposes Conditional class-Aware Meta-Learning (CAML) that incorporates class information by means of an embedding space to conditionally modulate representations of the base-learner. By conditionally transforming the intermediate representations of the base-learner, our goal is to reshape the representation with a global sense of class structure. Experiments reveal that the proposed conditional transformation can modulate the convolutional feature maps towards a more disentangled representation. We also introduce class-aware grouping to address a lack of statistical strength in few-shot learning. The proposed approach obtains comparable results with the current state-of-the-art performance on 5-way 1-shot *mini*ImageNet benchmark.

Chapter 5 introduces the medical imaging community to the rich field of meta-learning, which offers feasible solutions to the problem of limited training examples that the field is often faced with. To better evaluate realistic situations in the medical domain, we extend few-shot learning to medium-shot and establish a baseline procedure that aims to evaluate representative meta-learning algorithms on various amounts of training data. This serves as a baseline for future explorations using meta-learning in the medical domain. Through bias-variance analysis, we identify complementary roles of gradient-based and metric-based meta-learning and propose to fuse the best of both methods into Task Adaptive Metric Space. Our experiments reveal that the proposed metric adaptation method can adjust the metric space to better reflect examples of a new medical classification task.

Chapter 6 provides a summary of our findings and presents future research directions.

Chapter 2

Background and Related Work

2.1 Domain Adaptation

We follow the notations by Ben-David et al. [2010] and define a domain as an ordered pair consisting of a distribution $\mathcal{D}$ on the input space $\mathcal{X}$, and a labeling function $f:\mathcal{X}\rightarrow\mathcal{Y}$ that maps $\mathcal{X}$ to the label space $\mathcal{Y}$. The source and target domains are denoted by $\langle\mathcal{D}_S,f_S\rangle$ and $\langle\mathcal{D}_T,f_T\rangle$, respectively.

In unsupervised domain adaptation, the model is trained on labeled data from the source domain, together with unlabeled data from the target domain. The goal is to obtain a model $h\in\mathcal{H}$ which learns domain-invariant representations while simultaneously minimizing the classification error on $\mathcal{D}_S$.

2.1.1 A Brief Theory of Domain Adaptation

The bound on target domain error can be decomposed into source error ϵ_{S} and the divergence between the two distributions $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{T}}$.

f-divergence

A natural measure of domain divergence is the f-divergence, which measures the difference between two probability distributions P and Q.

$$D_f[P\,\|\,Q]=\int_{\Omega}f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right)\mathrm{d}Q \tag{2.1}$$

The f-divergence can be understood as an average, weighted by some function f, of the odds ratios given by P and Q.

The variation divergence, or total variation distance, is defined by the weight function

$f(u)=|u-1|$, in which case

$$D_f[P\,\|\,Q]=\int_{\Omega}f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}-1\right)\mathrm{d}Q=\int_{\Omega}\left(\frac{\mathrm{d}P-\mathrm{d}Q}{\mathrm{d}Q}\right)\mathrm{d}Q \tag{2.2}$$

$$=\int_{\Omega}(\mathrm{d}P-\mathrm{d}Q) \tag{2.3}$$

However, this cannot be measured accurately in the finite sample setting and it inflates the bound as it considers the σ-algebra [Ash et al., 2000], i.e., all subsets Ω of the input space, but we only care about a very small subset of all possible inputs spaces that is meaningful to our application. Furthermore, in the distribution-free setting, f-divergence can not give accurate estimates from finite samples. Instead, we use classifier-induced divergence from a hypothesis space $\mathcal{H}$.

$\mathcal{H}$-divergence

The $\mathcal{H}$-divergence [Ben-David et al., 2010] is defined as

$$d_{\mathcal{H}}(\mathcal{D},\mathcal{D}')=2\sup_{h\in\mathcal{H}}|P_{\mathcal{D}}[I(h)]-P_{\mathcal{D}'}[I(h)]| \tag{2.4}$$

where $I(h)$ is the characteristic (indicator) function of the set, i.e., $x\in I(h)$ when $h(x)=1$. The $\mathcal{H}$-divergence measures the maximum variation of two distributions over a subset on some function $h\in\mathcal{H}$. It is related to a *domain discriminator* that aims to provide an estimation of $\mathcal{H}$-divergence when all examples in $\mathcal{D}$ are predicted as 1, and all examples in $\mathcal{D}'$ are predicted as 0.

$\mathcal{H}\Delta\mathcal{H}$ divergence

To obtain errors bound of the target domain from the source domain, we need relative measures of divergence, i.e., h vs. h'. This will be used in Theorem 3.2.2 as a proxy to calculate the exact bound based on the optimal hypothesis h^* for each domain.

The *symmetric difference hypothesis space* $\mathcal{H}\Delta\mathcal{H}$ [Ben-David et al., 2010] is the set of disagreements between the two hypothesis h and h'

$$g\in\mathcal{H}\Delta\mathcal{H}\Leftrightarrow g(x)=h(x)\oplus h'(x) \tag{2.5}$$

The symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$ has the following property:

$$\epsilon_S(h,h')-\epsilon_T(h,h')\leq\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T), \tag{2.6}$$

where $\epsilon_S(h,h')$ denotes the disagreements between $h(x)$ and $h'(x)$ on data $\mathcal{D}_S$, and $\epsilon_T(h,h')$ denotes the disagreements between $h(x)$ and $h'(x)$ on data $\mathcal{D}_T$.

This is because

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) \tag{2.7}$$

$$=2\sup_{h,h'\in\mathcal{H}}|P_{x\sim\mathcal{D}_S}[h(x)\neq h'(x)]-P_{x\sim\mathcal{D}_T}[h(x)\neq h'(x)]| \tag{2.8}$$

$$=2\sup_{h,h'\in\mathcal{H}}|\epsilon_S(h,h')-\epsilon_T(h,h')| \tag{2.9}$$

$$\geq|\epsilon_S(h,h')-\epsilon_T(h,h')| \tag{2.10}$$

Theorem 2.1.1 (Bound of the target error [Ben-David et al., 2010]). *Let $\mathcal{H}$ be a hypothesis space of VC dimension d. $\mathcal{U}_S$, $\mathcal{U}_T$ are unlabeled samples of size m' each, drawn from $\mathcal{D}_S$, $\mathcal{D}_T$, respectively. For any $\delta\in(0,1)$, with probability at least $1-\delta$ (over the choice of samples), for every $h\in\mathcal{H}$:*

$$\epsilon_T(h) \tag{2.11}$$

$$\leq\epsilon_S(h)+\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S,\mathcal{U}_T)+4\sqrt{\frac{2d\log(2m')+\log(\frac{2}{\delta})}{m'}}+\lambda \tag{2.12}$$

where λ is the ideal joint expected loss, such that $\lambda=\epsilon_S(h^*)+\epsilon_S(h^*)$ from the hypothesis space $\mathcal{H}$. λ is often assumed to be negligible when the hypothesis space $\mathcal{H}$ is large enough. If λ is large, however, it means the hypothesis space does not contain a good hypothesis $h\in\mathcal{H}$ with small errors on both domains in which domain adaptations are not helpful.

2.1.2 Adversarial Learning for Domain Adaptation

Adversarial training is the prevailing approach for domain adaptation [Ganin et al., 2016]. It formulates a minimax problem where the maximizer maximizes the estimation of the domain divergence between the empirical samples, and the minimizer minimizes the sum of the source error and the domain divergence estimation obtained from the maximizer. The estimation of domain divergence can take many different forms, such as using a domain discriminator that predicts binary outcomes about whether the data is coming from the source or target distribution [Ganin et al., 2016], and estimating the discrepancies of two classifiers [Saito et al., 2018; Zhang et al., 2019b].

While matching the marginal distribution is a good step towards domain-invariant learning, it is still susceptible to the problem of conditional distribution mismatching. Prototype-based class-conditioned domain alignment [Chen et al., 2019a; Liang et al., 2019a,b; Luo et al., 2017; Pan et al., 2019; Xie et al., 2018] is designed to address this problem. We refer to this group of methods as *explicit* class-conditioned domain alignment. The explicit alignment is achieved by incorporating an auxiliary loss that minimizes the Euclidean distance of the class-conditioned prototypical representations $\mathbf{c}_j$ between the source and target domains. The class-conditioned prototype $\mathbf{c}_j$ is the average representation for all examples in a domain with class label j.

The main limitation of explicit class-conditioned domain alignment is in its reliance on explicit optimization of model parameters based on model's predictions on the target domain as pseudo-labels. This learning procedure is vulnerable to error accumulation [Chen et al., 2019a] as mistakes in the pseudo-label predictions can gradually accumulate leading to poor local minima in EM-style training. Furthermore, the pseudo-labels are likely to suffer from ill-calibrated probabilities [Guo et al., 2017], especially for deep learning methods, which exacerbate the critical problem of error accumulation with misleadingly confident mistakes.

2.1.3 Related Work for Domain Adaptation

We review related work on unsupervised domain adaptation and discuss their relationship to our proposed method.

Instance-based importance-weighting [Chawla et al., 2002; Kouw and Loog, 2019] aims to minimize the target error directly from the source domain data, weighted at the example level or class level. Example-level weighting is designed to address covariate shift that uses important-weighting to train the classifier from the source domain data. It first estimates of the probability of a source example belonging to the target domain, then uses that probability as important-weighting to train the classifier from the source domain data. Conversely, class-level weighting applies the weighting on class labels and is designed to address cost-sensitive learning [Elkan, 2001] and class imbalance. Oversampling methods [Chawla et al., 2002] have an equivalent effect with importance-weighting. Unlike our approach, importance-weighting only uses the source data to train the classifier without learning domain invariant representations.

Feature-based distribution adaptation is the prevailing approach to domain adaptation that aims to minimize the distribution discrepancy between the source and target domains. The domain difference can be measured in various ways, such as Maximum Mean Discrepancy (MMD) [Borgwardt et al., 2006], which is further minimized to achieve domain invariance. The minimization of such discrepancy can be carried out by directly minimizing the distance [Tzeng et al., 2014] or with the help of adversarial learning [Ganin et al., 2016].

Classifier-based distribution adaptation is a strong competitor to feature-based adaptation. It aims to minimize the discrepancy between two classifiers so that the learned representations respect the decision boundary of the classification task [Saito et al., 2018; Zhang et al., 2019b]. We use classifier-based discrepancy MDD for adversarial training because the probabilities predicted by two classifiers are more informative than domain discriminator-based binary outcomes.

Feature-classifier joint distribution adaptation aims to align the joint distribution between features and their corresponding predictions [Long et al., 2013; Tsai et al., 2018]. The joint distribution can be represented in a multilinear map between features and classifier predictions [Long et al., 2018], or the Cartesian product between the domain space and class space [Cicek and Soatto, 2019]. In our work, we implicitly align the joint distribution with the factorization $p(x,y) = p(x|y)p(y)$ from a sampling perspective where $p(y)$ is the pre-specified alignment distribution in the label space, and $p(x|y)$ represents class-conditioned sampling.

Explicit class-conditioned domain alignment, or class prototype alignment, introduces a loss function that minimizes the distances of class-level prototypes between the source and target domains [Deng et al., 2019; Pan et al., 2019; Pinheiro, 2018; Snell et al., 2017a]. It is prone to error accumulation due to its reliance on explicit optimization of model parameters from the pseudo-labels. A variety of recent methods have been proposed to mitigate these limitations by estimating batch-level statistics [Xie et al., 2018] and introducing an easy-to-hard curriculum that favors confident predictions [Chen et al., 2019a]. Nevertheless, these algorithms suffer from ill-calibrated probabilities in the form of confident mistakes, and more work is needed to improve model calibration so as to better utilize explicit alignment.

Self-training [Nigam and Ghani, 2000] is a special form of co-training [Blum and

Mitchell, 1998] where the model iteratively uses its predictions on the unlabeled examples, i.e., pseudo-labels, as explicit supervision to re-train itself. The use of pseudo-labels has become an emerging trend in domain adaptation, because they provide estimations of the target domain label distribution that can be exploited by training algorithms. Apart from class prototype based methods [Chen et al., 2011; Deng et al., 2019; Saito et al., 2017; Zhang et al., 2018] for explicit alignment, [Wen et al., 2019] proposed the use of uncertainty estimates of the target domain predictions as second-order statistics to promote feature-label joint adaptation. For semantic segmentation tasks, [Zou et al., 2018] proposed to iteratively generate pseudo-labels in the target domain and re-train the model on these labels; [Zhang et al., 2019a] proposed to use pseudo-labels to encourage examples to cluster together if they belong to the same class; [Chen et al., 2019b] applied entropy minimization [Grandvalet and Bengio, 2005] on the pseudo-labels to encourage class overlap between domains. A main bottleneck for this approach is the bias in pseudo-label predictions. Directly optimizing these labels is prone to "entropy over-minimization" [Zou et al., 2019] and negative transfer [Lifshitz and Wolf, 2020] where the model overfits to mistakes in the pseudo-labels. Moreover, the pseudo-labels are likely to suffer from ill-calibrated probabilities [Guo et al., 2017], especially for deep learning methods, where examples needs to be chosen such that they adhere to their predictive uncertainties. The resulting misleadingly confident mistakes exacerbate the critical problem of error accumulation in pseudo-label bias. In contrast, our proposed method removes the need for direct supervision from pseudo-labels, and as a result is more robust to bias in how these labels are produced.

Reinforced sample selection [Dong and Xing, 2018] is proposed in the one-shot domain adaptation setup where the model actively selects labeled examples in the source domain conditioned on the target domain examples for more accurate distance-based metric learning.

2.2 Meta-learning

Meta-learning [Bengio et al., 1992; Branco et al., 2018; Ling and Sheng, 2010; Mitchell and Thrun, 1993; Schmidhuber, 1987; Vilalta and Drissi, 2002] has been studied extensively in the machine learning literature. Although "different researchers hold different views of what the term meta-learning exactly means", the common goal of meta-learning is to "exploit the knowledge of learning (meta-knowledge) to improve the performance of learning algorithms" [Vilalta and Drissi, 2002]*. The meta-knowledge can be manifested in many different ways, such as predictions of base-learners in stacked generalization [Freund et al., 1996; Wolpert, 1992].

In deep learning, meta-learning has been studied as a means to acquire meta-knowledge across many tasks. In contrast to standard supervised learning setup where the goal is to learn a single model from a task $\mathcal{D}$, a meta-learner aims to learn from a set of tasks $\{\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_S\}$ to get better at learning new tasks. At the meta-level, each task $\mathcal{D}_i$ is treated as a training example. It is known as meta-learning so as to improve a model's learning ability when the meta-learner is exposed to more and more tasks over time.

In recent years, meta-learning has become an important approach for few-shot learning. Previous work on deep learning based meta-learning approaches can be summarized as: learning representations that encourage fast adaptation on new tasks [Finn et al., 2017a,b], learning universal learning procedure approximators by supplying training examples to the meta-learner that outputs predictions on the testing examples [Hochreiter et al., 2001; Mishra et al., 2017; Santoro et al., 2016; Vinyals et al., 2016], learning to generate model parameters conditioned on training examples [Gomez and Schmidhuber, 2005; Ha et al., 2016; Munkhdalai and Yu, 2017], learning optimization algorithms [Andrychowicz et al., 2016; Bengio et al., 1992; Li and Malik, 2017; Ravi and Larochelle, 2016], and learning a metric space for distance-based inference [Oreshkin et al., 2018; Ren et al., 2018; Snell et al., 2017b].

This section describes the meta-learning problem formulation [Ravi and Larochelle, 2016] and revisits the gradient-based and metric-based meta-learning methods.

2.2.1 Meta-learning Problem Formulation

In a traditional classification setup, there exists a classification task $\mathcal{D}$ which contains M number of classes. The examples of this task are typically divided into a training set $\mathcal{D}^{\text{train}}$, a validation set $\mathcal{D}^{\text{valid}}$ and a test set $\mathcal{D}^{\text{test}}$. An M-way classifier is trained on

*In this thesis, the term "meta-learning" is restricted to deep learning based approaches for learning across different tasks.

$\mathcal{D}^{\text{train}}$ to optimize for parameters θ that minimize some loss $\mathcal{L}$,

$$\theta^{*} = \text{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}^{\text{train}};\theta). \tag{2.13}$$

The validation set $\mathcal{D}^{\text{valid}}$ is used for hyperparameter selection and the generalization error of the model is estimated on test set $\mathcal{D}^{\text{test}}$.

In a meta-learning setup, however, the goal is to learn from a *distribution of tasks*. The learning happens on two levels: (i) a meta-level model, or meta-learner, that learns across many tasks, and (ii) a base-level model, or base-learner, that operates within each specific task. Meta-learning happens in task space, where each task can be treated as one meta-example. In the meta-learning formulation, we define a collection of regular tasks as meta-sets $\mathscr{D}$, and each task $\mathcal{D} \in \mathscr{D}$ has its own $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ split. $\mathcal{D}^{\text{train}}$ is often denoted as the "support set" and $\mathcal{D}^{\text{test}}$ the "query set". The resulting meta-learner objective is to choose parameters θ that minimize the expected loss $\mathcal{L}(\cdot;\theta)$ across all tasks in $\mathscr{D}$,

$$\theta^{*} = \text{argmin}_{\theta} \mathbb{E}_{\mathcal{D} \sim \mathscr{D}}[\mathcal{L}(\mathcal{D};\theta)].$$

At the meta-level, the meta-sets $\mathscr{D}$ can be further split into disjoint meta-training set $\mathscr{D}_{\text{meta}-\text{train}}$, meta-validation set $\mathscr{D}_{\text{meta}-\text{valid}}$ and meta-test set $\mathscr{D}_{\text{meta}-\text{test}}$. The meta-learner is trained on $\mathscr{D}_{\text{meta}-\text{train}}$, validated on $\mathscr{D}_{\text{meta}-\text{valid}}$ and finally evaluated on $\mathscr{D}_{\text{meta}-\text{test}}$. In each meta-training iteration, we sample a batch of tasks from $\mathscr{D}_{\text{meta}-\text{train}}$ and use gradient-based methods to optimize the meta-learner. This is akin to stochastic gradient descent at the meta-level task space.

Relation with transfer learning and multitask learning

Transfer learning can be broadly understood as improving the performance of a learner by transferring the knowledge from related tasks or learning systems. Inductive transfer tends to be used interchangeably with transfer learning, where the notion of representation transfer in deep learning typically refers to training a model on a larger dataset, such as ImageNet, and transfer it to a much smaller dataset. Meta-learning and multitask learning [Caruana, 1997; Silver et al., 2008] are different approaches to achieve inductive transfer by learning different tasks simultaneously. Multitask learning shares inductive biases across tasks by learning all tasks in parallel while using a shared representation. On the other hand, meta-learning shares inductive bias in a meta-learner that imposes dynamic bias to each task. As a consequence, multitask learning is preferable when the task of interest is readily available together with many related tasks. In contrast, meta-learning is preferable if the goal is to generalize to a future task with few examples and the task is not currently available. However, it is possible to address a meta-learning problem with multitask learning approaches, such as inferring task representation and use it in context-sensitive multitask learning. Transductive transfer is a parallel technique with inductive transfer pioneered by Gammerman et al. [1998]. The main distinction with inductive transfer is that, instead of transferring a learned model to another model on a new dataset, it might be easier to infer predictions on the new dataset before obtaining a new model. Unsupervised domain adaptation can be considered a type of transductive transfer.

2.2.2 Related Work for Deep Learning- Based Meta-learning

Gradient-Based Approaches

Model-Agnostic Meta-learning [Finn et al., 2017a] is a gradient-based meta-learning algorithm that aims to learn representations that encourage fast adaptation across different tasks. The meta-learner and base-learner share the same network structure, and the parameters learned by the meta-learner are used to initialize the base-learner on any given task.

To form an “episode” [Vinyals et al., 2016] to optimize the meta-learner, we first sample a set of tasks $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_S\}$ from the meta-training set $\mathscr{D}_{\text{meta-train}}$, where $\mathcal{D}_i = \{\mathcal{D}_i^{\text{train}}, \mathcal{D}_i^{\text{test}}\}$. For a meta-learner parameterized by θ, we compute its adapted parameters θ_i for each sampled task $\mathcal{D}_i$:

$$\theta_i \leftarrow \theta - \beta_{\text{T}} \nabla_\theta \mathcal{L}(\mathcal{D}_i^{\text{train}}; \theta), \tag{2.14}$$

where β_{T} is the step size of the gradient. The adapted parameters θ_i are task-specific and tell us the effectiveness of θ as to whether it can achieve generalization through one or a few additional gradient steps. The meta-learner's objective is hence to minimize the generalization error of θ across all tasks:

$$\theta^* = \text{argmin}_\theta \sum_{\mathcal{D}_i \sim \mathscr{D}_{\text{meta-train}}} \mathcal{L}(\mathcal{D}_i^{\text{test}}; \theta_i). \tag{2.15}$$

Note that the meta-learner is not aimed at explicitly optimizing the task-specific parameters θ_i. Rather, the objective of the meta-learner is to optimize the representation θ so that it can lead to good task-specific adaptations θ_i with a few gradient steps. In other words, the goal of fast learning is integrated into the meta-learner's objective.

The meta-learner is optimized by backpropagating the error through the task-specific parameters θ_i to their common pre-update parameters θ. The gradient-based updating rule is:

$$\theta \leftarrow \theta - \beta_{\mathrm{M}} \nabla_\theta \sum_{\mathcal{D}_i \sim \mathscr{D}_{\mathrm{meta-train}}} \mathcal{L}(\mathcal{D}_i^{\mathrm{test}}; \theta_i), \tag{2.16}$$

where β_{M} is the learning rate of the meta-learner. The meta-learner performs slow learning at the meta-level across many tasks to support fast learning on new tasks. At meta-test time, we initialize the base-learner's parameters from the meta-learned representation θ^* and fine-tune the base-learner using gradient descent on task $\mathcal{D}_i^{\mathrm{train}} \sim \mathscr{D}_{\mathrm{meta-test}}$. The meta learner is evaluated on $\mathcal{D}_i^{\mathrm{test}} \sim \mathscr{D}_{\mathrm{meta-test}}$.

MAML works with any differentiable neural network structure and has been applied to various tasks including regression, image classification, reinforcement learning and imitation learning. Extensions of MAML include learning the base-learner's learning rate [Li et al., 2017] and applying a bias transformation to concatenate a vector of parameters to the hidden layer of the base-learner [Finn et al., 2017b]. It is also theorized that MAML has the same expressive power as other universal learning procedure approximators and generalizes well to out-of-distribution tasks [Finn and Levine, 2017].

Metric-Based Approaches

Siamese networks [Koch et al., 2015] learn a similarity measure between inputs using a shared network architecture that outputs high probability when paired examples are from the same class. Matching networks [Vinyals et al., 2016] use full context embeddings to encode examples to the metric space and use attention as a similarity measure for predictions.

Prototypical networks [Snell et al., 2017b] compute a centroid, or prototype, for every class that are later used for distance-based queries of new examples. Examples of the input space are encoded through a learned function f_ϕ in the form of an M-dimensional metric space, where each input example is reduced to a point in the metric space. The centroid $\mathbf{c}_t$ for each class t is defined as:

$$\mathbf{c}_t = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}^{\text{train}}} \mathbb{1}_{\{y_i = t\}} f_\phi(\mathbf{x}_i),$$

where K denotes the number of examples for class t, $\mathbb{1}_{\{y_i=t\}}$ denotes an indicator function of y_i which takes value 1 when $y_i = t$ and 0 otherwise. The mapping function f_ϕ is optimized to minimize the negative log-probability defined in Eq. (2.17) by minimizing the Euclidean distance d between an example and its corresponding class centroid $\mathbf{c}_t$ while maximizing its Euclidean distance to other class centroids $\mathbf{c}_{t'}$:

$$\underset{\phi}{\operatorname{argmin}} \mathbb{E} \left[d(f_\phi(\mathbf{x}_i), \mathbf{c}_t)) + \log \sum_{t'} \exp(-d(f_\phi(\mathbf{x}_i), \mathbf{c}_{t'})) \right]. \tag{2.17}$$

The goal of metric learning is to obtain the mapping function f_ϕ so that examples can be mapped from the input space to the metric space in a semanitcally meaningful way. The learned metric space has broad applications including image classification [Mensink et al., 2012], face recognition [Schroff et al., 2015], information retrieval and ranking [McFee and Lanckriet, 2010].

Chapter 3

Implicit Class-Conditioned Domain Alignment for Unsupervised Domain Adaptation

3.1 Introduction

Supervised learning aims to extract statistical patterns from data by learning to approximate the conditional density $p(y|x)$. However, the generalization of the approximation is often sensitive to some dataset-specific factors. Dataset shift [Quionero-Candela et al., 2009] frequently arises from real-world applications and can manifest in many different ways, such as sample selection bias [Heckman, 1979; Torralba et al., 2011], class distribution shift [Webb and Ting, 2005], and covariate shift [Shimodaira, 2000]. Unsupervised Domain Adaptation (UDA) aims to address domain shift with access to labeled data in the source domain and unlabeled data in the target domain [Pan and Yang, 2009]. The fundamental algorithmic issue is to infer domain-invariant representations.

While considerable progress has been made in UDA [Ganin et al., 2016], they tend to focus on marginal distribution matching in the feature space, and less emphasis is made on discovering label distributions. In real-world applications, it is very common to have *class imbalance* [Chawla, 2009; Japkowicz and Stephen, 2002] within each domain and *class distribution shift* [Tan et al., 2019] between different domains, necessitating the incorporation of label space distribution into adaptation. *Explicit* class-conditioned domain alignment [Deng et al., 2019; Liang et al., 2019a; Pan et al., 2019; Xie et al., 2018] has emerged as a key approach to promoting class-conditioned invariance by aligning prototypical representations of each class. While explicit alignment has the advantage of directly minimizing class-conditioned misalignment, it presents critical vulnerabilities to error accumulation [Chen et al., 2019a] and ill-calibrated probabilities [Guo et al., 2017] due to its dependence on *explicit* supervision from pseudo-labels provided by model predictions.

We propose *Implicit* Class-Conditioned Domain Alignment that removes the need for explicit pseudo-label based optimization. Instead, we use the pseudo-labels *implicitly* to *sample* class-conditioned data in a way that maximally aligns the joint distribution between features and labels. The primary advantage of the sampling-based implicit domain alignment is the ability to address between-domain class distribution shift by imposing an uniform class distribution between the source and target domains. The proposed method is simple, effective, and is supported by theoretical analysis on the empirical estimations of domain divergence measures. It also overcomes limitations of explicit alignment by allowing the domain adaptation algorithm to discover class-conditioned domain-invariance in an unsupervised way without explicit supervision from pseudo-labels. Note that the proposed approach does not address class imbalance; instead, it assumes the uniform cost for different misclassification errors.

The contributions of this chapter are as follows: (i) We propose implicit class-conditioned domain alignment to address between-domain class distribution shift, which also overcomes the limitations of explicit domain alignment; (ii) We provide a theoretical analysis by decomposing the empirical domain divergence into class-aligned and class-misaligned divergence, and show that class-misaligned divergence is detrimental to domain adaptation; (iii) We report state-of-the-art UDA performance under extreme between-domain class distribution shift, as well as competitive results on standard UDA tasks; (iv) We show that method offers consistent improvements for different adversarial adaptation algorithms: both DANN and .

3.2 Method

We begin with theoretical motivations of implicit alignment by decomposing the empirical domain divergence measure into class-aligned and class-misaligned divergence, and show that misaligned divergence is detrimental to domain adaptation. We then present the proposed implicit domain alignment framework that addresses class-misalignment.

3.2.1 Theoretical Motivations

The $\mathcal{H}\Delta\mathcal{H}$ divergence between two domains is defined as

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T)=2\sup_{h,h'\in\mathcal{H}}|\mathbb{E}_{\mathcal{D}_T}[h\neq h']-\mathbb{E}_{\mathcal{D}_S}[h\neq h']|, \tag{3.1}$$

where $\mathcal{H}$ denotes some hypothesis space, and $h\neq h'$ is the abbreviation for $h(x)\neq h'(x)$. [Ben-David et al., 2010] theorized that the target domain error $\epsilon_T(h)$ is bounded by the error of the source domain $\epsilon_S(h)$ and the empirical domain divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S,\mathcal{U}_T)$ where $\mathcal{U}_S$, $\mathcal{U}_T$ are unlabeled empirical samples drawn from $\mathcal{D}_S$, $\mathcal{D}_T$.

In deep learning, minibatch-based optimization limits the amount of data available at each training step. This necessitates the analysis of the empirical estimations of $d_{\mathcal{H}\Delta\mathcal{H}}$ at the minibatch level, so as to shed light on the learning dynamics. Implicit domain alignment plays an important role in the empirical estimations of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T)$, where only finite samples $\mathcal{U}_S$, $\mathcal{U}_T$ are available. Below we define the empirical $\mathcal{H}\Delta\mathcal{H}$ divergence on mini-batches.

Definition 3.2.1. *Let $\mathcal{B}_S$, $\mathcal{B}_T$ be minibatches from $\mathcal{U}_S$ and $\mathcal{U}_T$, respectively, where $\mathcal{B}_S \subseteq \mathcal{U}_S$, $\mathcal{B}_T \subseteq \mathcal{U}_T$, and $|\mathcal{B}_S| = |\mathcal{B}_T|$. The empirical estimation of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S,\mathcal{B}_T)$ over the minibatches $\mathcal{B}_S$, $\mathcal{B}_T$ is defined as*

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S,\mathcal{B}_T) = \sup_{h,h'\in\mathcal{H}} \left| \sum_{\mathcal{B}_T}[h \neq h'] - \sum_{\mathcal{B}_S}[h \neq h'] \right|. \tag{3.2}$$

Theorem 3.2.2 (The decomposition of $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S,\mathcal{B}_T)$)**.** *Let $\mathcal{H}$ be a hypothesis space and $\mathcal{Y}$ be the label space of the classification task where $\mathcal{B}_S$, $\mathcal{B}_T$ are minibatches drawn from $\mathcal{U}_S$, $\mathcal{U}_T$, respectively, and Y_S, Y_T are the label set of $\mathcal{B}_S$, $\mathcal{B}_T$. We define three disjoint sets on the label space: the shared labels $Y_C := Y_S \cap Y_T$, and the domain-specific labels $\overline{Y_S} := Y_S - Y_C$, and $\overline{Y_T} := Y_T - Y_C$. We also define the following disjoint sets on the input space where $\mathcal{B}_S^C := \{x \in \mathcal{B}_S \,|\, y \in Y_C\}$, $\mathcal{B}_S^{\overline{C}} := \{x \in \mathcal{B}_S \,|\, y \notin Y_C\}$, $\mathcal{B}_T^C := \{x \in \mathcal{B}_T \,|\, y \in Y_C\}$, $\mathcal{B}_T^{\overline{C}} := \{x \in \mathcal{B}_T \,|\, y \notin Y_C\}$. The empirical $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S,\mathcal{B}_T)$ divergence can be decomposed into class aligned divergence and class-misaligned divergence:*

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S,\mathcal{B}_T) = \sup_{h,h'\in\mathcal{H}} \left| \xi^C(h,h') + \xi^{\overline{C}}(h,h') \right|, \tag{3.3}$$

where

$$\xi^C(h,h') = \sum_{\mathcal{B}_T^C} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^C} \mathbb{1}[h \neq h'], \tag{3.4}$$

$$\xi^{\overline{C}}(h,h') = \sum_{\mathcal{B}_T^{\overline{C}}} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^{\overline{C}}} \mathbb{1}[h \neq h']. \tag{3.5}$$

Proof. By definition, we have

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{B}_S,\mathcal{B}_T) = \sup_{h,h'\in\mathcal{H}} \left| \sum_{\mathcal{B}_T} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S} \mathbb{1}[h \neq h'] \right| \tag{3.6}$$

We rewrite the summation over all the samples $\mathcal{B}$ into the sum of disjoint subsets $\mathcal{B}^C$ and $\mathcal{B}^{\overline{C}}$.

$$\sum_{\mathcal{B}_T} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S} \mathbb{1}[h \neq h'] \tag{3.7}$$

$$= \left(\sum_{\mathcal{B}_T^C} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^C} \mathbb{1}[h \neq h'] \right) \tag{3.8}$$

$$+ \left(\sum_{\mathcal{B}_T^{\overline{C}}} \mathbb{1}[h \neq h'] - \sum_{\mathcal{B}_S^{\overline{C}}} \mathbb{1}[h \neq h'] \right) \tag{3.9}$$

$$= \xi^C(h,h') + \xi^{\overline{C}}(h,h'). \tag{3.10}$$

This completes the proof. □

Remark 3.2.3 (The domain discriminator shortcut)**.** *Let the ordered triple (x, y_c, y_d) denote data sample x, and its associating class label y_c and domain label y_d, respectively, where $x \in \mathcal{B}$, $y_c \in Y$ and $y_d \in \{0,1\}$. Let f_c be a classifier that maps x to a class label y_c. Let f_d be a domain discriminator that maps x to a binary domain label y_d. For the empirical class-misaligned divergence $\xi^{\overline{C}}(h,h')$ with sample $x \in \mathcal{B}_S^{\overline{C}} \cup \mathcal{B}_T^{\overline{C}}$, there exists a domain discriminator shortcut function*

$$f_d(x) = \begin{cases} 1 & f_c(x) \in \overline{Y_S} \\ 0 & f_c(x) \in \overline{Y_T}, \end{cases} \tag{3.11}$$

such that the domain label can be solely determined by the domain-specific class labels. This shortcut interferes with adversarial domain adaptation because the model could bypass the optimization for domain-invariant representations, but rather optimize for a shortcut function that is independent of the covariate contributing to the domain difference.

Figure 3.1 illustrates a toy example where the source and target domains are aligned for class 4 but misaligned between classes 3 and 6 as a result of random sampling in the minibatch construction. The domain discriminator aims to predict domain labels based on their domain information, i.e., red and blue. However, due to the class shortcut for the misaligned samples (3 and 6), the domain discriminator could infer domain labels based on class information directly (digits 3 and 6), without the need to learn domain-specific

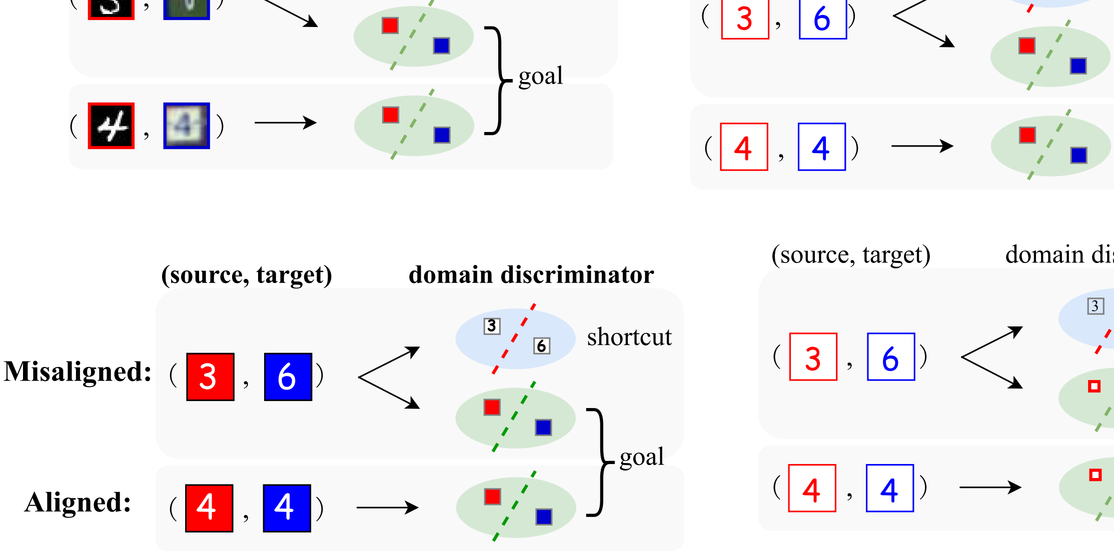


Figure 3.1: Illustration of the domain discriminator shortcut. The domain discriminator aims to distinguish between different domains (red and blue), where the decision boundary is represented by dashed lines. But misaligned samples create a shortcut where the domain labels can be directly determined by the misaligned class labels (3 and 6). The decision boundary of the resulting shortcut is independent of the covariate that causes the domain difference, which does not contribute to adversarial domain-invariant learning.

information. This problem of class-misalignment is especially pronounced under extreme between-domain class distribution shift, where a simple random sample is more likely to fail in providing good coverage of the label space.

3.2.2 Implicit Class-Conditioned Domain Alignment

Having identified the domain discriminator shortcut in class misaligned empirical samples, we now propose framework that aligns the two domains from a sampling perspective, providing a unified adversarial training procedure without the use of additional losses or hyper-parameters.

Figure 3.2 depicts the proposed implicit class-conditioned domain alignment framework. We aim to align $p_S(x)$ and $p_T(x)$ at the input and label space jointly with the factorization $p(x,y) = p(x|y)p(y)$ while ensuring that the sampled classes are aligned between the two domains. The alignment distribution $p(y)$ is pre-specified, e.g., uniform distribution, to ensure samples are aligned in the shared label space in spite of different empirical label distributions of the two domains. This implicit alignment procedure minimizes the class-misaligned divergence $\xi^{\overline{C}}(h,h')$, providing a more reliable empirical estimation of domain divergence. For the unlabeled target domain, we use the model predictions to sample class-conditioned data from $p_T(x|\hat{y})$ to approximate $p_T(x|y)$.

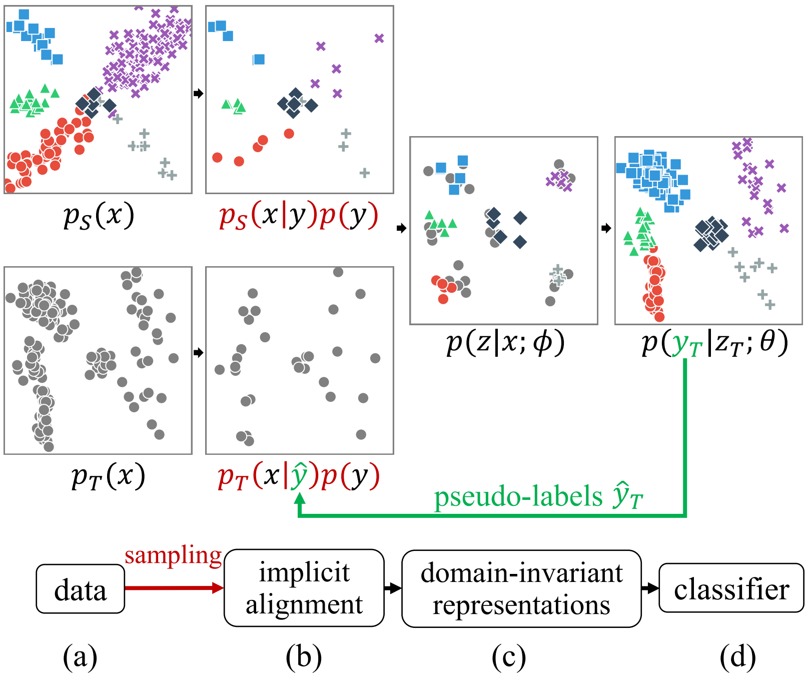


Figure 3.2: The proposed framework. (a) We aim to align the source domain $p_S(x)$, colored by classes, with unlabeled target domain $p_T(x)$. (b) For $p_S(x)$, we sample $x \sim p_S(x|y)p(y)$ based on the *alignment distribution* $p(y)$. For $p_T(x)$, we sample a *class aligned* minibatch $x \sim p_T(x|\hat{y})p(y)$ using identical $p(y)$, with the help of pseudo-labels $\hat{y}_T$. (c) The adversarial training aims to acquire domain-invariant representations z from the feature extractor parameterized by ϕ. (d) The classifier predicts class labels from z.

3.2.3 Class-Aligned Sampling Strategy

Algorithm 1 presents the proposed sampling procedure that selects class-aligned examples for minibatch training. It is a type of stratified sampling where the dataset is partitioned into mutually exclusive subgroups to reflect the label information in a class-aligned manner.

First, we predict pseudo-labels of the target domain using the classifier $f_c(\cdot;\theta)$ parameterized by θ. The pseudo-labels will be later used in class-conditioned sampling. Second, we sample a set Y from the label space $\mathcal{Y}$ where $p(y)$ defines the probability with which we pick the classes to align so as to ensure the empirical samples of the source and target domains share the same Y. This in turn minimizes the class-misaligned divergence $\xi^{\overline{C}}(h,h')$. Third, for each class $y_i \in Y$, we sample class-conditioned examples for the source and target domains, respectively, and store them in (X'_S,Y'_S) and X'_T. This is equivalent to performing a table lookup to select a subset $\mathcal{B}_i$ where all examples belong to class y_i, followed by random sampling in $\mathcal{B}_i$. We use pseudo-labels to sample the

Algorithm 1 The proposed implicit alignment training

```
Input: dataset $S=\{(x_i,y_i)\}_{i=1}^N$, $T=\{x_i\}_{i=1}^M$,
       label space $\mathcal{Y}$, label alignment distribution $p(y)$, classifier $f_c(\cdot;\theta)$
while not converged do
    # predict pseudo-labels for $T$
    $\hat{T} \leftarrow \{(x_i, f_c(x_i;\theta))\}_{i=1}^M$ where $x_i \in T$
    # sample $N$ unique classes in the label space
    $Y \leftarrow$ draw $N$ samples in $\mathcal{Y}$ from $p(y)$
    # sample $K$ examples conditioned on each $y_i \in Y$
    for $y_i$ in $Y$ do
        $(X'_S, Y'_S) \leftarrow$draw $K$ samples in $S$ from $p_S(x|y_i)$
        $X'_T \leftarrow$draw $K$ samples in $\hat{T}$ from $p_T(x|y_i)$
    end for
    # domain adaptation training on this minibatch
    train minibatch $(X'_S, Y'_S, X'_T)$
end while
```

target domain due to the lack of ground-truth labels. Once we obtained the class-aligned minibatch, we use it to train unsupervised domain adaptation algorithm and repeat this process until the model converges.

This algorithm addresses class distribution shift between different domains by specifying the sampling strategy $p(y)$ in the label space. We use uniform sampling $p(y)$ for all experiments in this chapter, and leave more advanced specifications and their applications to cost-sensitive domain adaptation as future work.

For the worst-case analysis, given that models do not initially perform well when training begins, for a random classifier, implicit alignment selects random samples that is equivalent to the vanilla stochastic batch training strategy without any sampling. Although errors could inevitably be present in pseudo-labels, the proposed approach suffers from the pseudo-labels bias to a lesser degree than the explicit alignment approach—the pseudo-labels are only used in sampling. Implicit alignment does not optimize the model explicitly towards its predictions.

3.2.4 Integrating Implicit Alignment into Classifier-Based Domain Discrepancy Measure

Section 3.2.3 describes the implicit alignment algorithm from a sampling perspective, where we sample minibatches in a way that maximizes class alignment implicitly. This sampling strategy is independent of the choice of domain divergence measures. In this section, we show how to integrate the sampling approach into Margin Disparity Discrepancy (MDD) [Zhang et al., 2019b]—a state-of-the-art classifier-based domain discrepancy measure—to further facilitate implicit alignment. MDD is defined as

$$d_{f,\mathcal{F}}(S,T) = \sup_{f' \in \mathcal{F}} \Big(\mathrm{disp}_{\mathcal{D}_T}(f',f) - \mathrm{disp}_{\mathcal{D}_S}(f',f) \Big), \tag{3.12}$$

where f and f' are two independent scoring functions that predict class probabilities, and $\mathrm{disp}(f',f)$ is a disparity measure between the scores provided by the classifiers f' and f. The domain divergence is to estimate the discrepancy between the disparity measures of the two domains.

Following notations in Theorem 3.2.2, we define the empirical MDD on class-misaligned samples as

$$\hat{d}_{f,\mathcal{F}}(\mathcal{B}_S^{\overline{C}},\mathcal{B}_T^{\overline{C}}) = \sup_{f' \in \mathcal{F}} \Big(\sum_{\mathcal{B}_T^{\overline{C}}} \mathrm{disp}(f',f) - \sum_{\mathcal{B}_S^{\overline{C}}} \mathrm{disp}(f',f) \Big). \tag{3.13}$$

Because $\mathcal{B}_S^{\overline{C}}$ and $\mathcal{B}_T^{\overline{C}}$ are disjoint in the label space, there exists a shortcut solution

$$\mathrm{disp}(f'(x),f(x)) = \begin{cases} 0 & f_c(x) \in \overline{Y_S} \\ 1 & f_c(x) \in \overline{Y_T}, \end{cases} \tag{3.14}$$

which maximizes the divergence estimation of Eq. (3.13). Although class-aligned sampling can mitigate this problem, it is difficult to fully eliminate the impact of misalignement due to imperfect pseudo-labels. To further eliminate the detrimental impact of class-misalignment, we introduce a masking scheme on the scoring functions f and f' defined as

$$\begin{aligned} &\hat{d}_{f,\mathcal{F}}(\mathcal{B}_S,\mathcal{B}_T) \\ &= \sup_{f' \in \mathcal{F}} \Big(\sum_{\mathcal{B}_T} \mathrm{disp}(f' \odot \omega, f \odot \omega) - \sum_{\mathcal{B}_S} \mathrm{disp}(f' \odot \omega, f \odot \omega) \Big), \end{aligned} \tag{3.15}$$

where $f \odot \omega$ denotes element-wise multiplication between the output of f and ω. The alignment mask ω is a binary vector that denotes whether the i-th class is present in the sampled classes Y (i.e., the classes that we intend to align in the current minibatch). By doing so, we simultaneously align the source and target domains (i) in the input space and (ii) in the functional approximations of the domain divergence by masking the scoring functions f and f'.

3.3 Experiments on Standard Domain Adaptation Datasets

3.3.1 Setup

Datasets. We evaluate on Office-31 and Office-Home. Office-31 [Saenko et al., 2010] has three domains (**A**mazon, **D**SLR and **W**ebcam) with 31 classes. Office-Home [Venkateswara et al., 2017] contains four domains (**Ar**t, **Cl**ip Art, **Pr**duct, and **Re**al-world) with 65 classes. We use three versions of Office-Home [Venkateswara et al., 2017] that contains four domains (**Ar**t, **Cl**ip Art, **Pr**duct, and **R**eal-**w**orld) with 65 classes: (i) "standard": the standard Office-Home dataset. (ii) "balanced" [Tan et al., 2019]: a subset of the standard dataset where each class has the same number of examples. (iii) "RS-UT": Reversely-unbalanced Source (RS) and Unbalanced-Target (UT) distribution [Tan et al., 2019] where both domains are imbalanced, but the majority class in the source domain is the minority class in the target domain. Further dataset details are in the supplementary materials.

Model Details. We use ResNet-50 [He et al., 2016] pre-trained from ImageNet [Russakovsky et al., 2015] as the backbone, and use hyper-parameters from [Zhang et al., 2019b] for MDD-based domain discrepancy measure. The batch size is 31 for Office-31 and 50 for Office-Home. We use PyTorch 1.2 as training environments. The backbone is followed by a 1-1ayer bottleneck. The classifier f and auxiliary classifier f' are both 2-layer networks. We use the SGD optimizer with learning rate 0.001, nesterov momentum 0.9, and weight decay 0.0005. We empirically find that SGD converges better than Adam for adversarial optimization. We use a gradient reversal layer for minimax optimization, and we use a training scheduler [Ganin et al., 2016] for gradient reversal layer defined as

$$\lambda_p = \frac{0.2}{1+\exp(-\frac{i}{1000})} - 0.1, \tag{3.16}$$

where i denotes the step number. We used the same scheduler from [Zhang et al., 2019b] for all experiments and have not tried hyperparameter search for λ_p. For Office-Home, the learning rate for the ResNet-50 backbone is 0.0001 and the learning rate for the remaining parameters are 0.001. For Office-31, the learning rate for the ResNet-50 backbone is 0.001 and the learning rate for the remaining parameters are 0.01. The batch size is 31 for Office-31 and 50 for Office-Home.

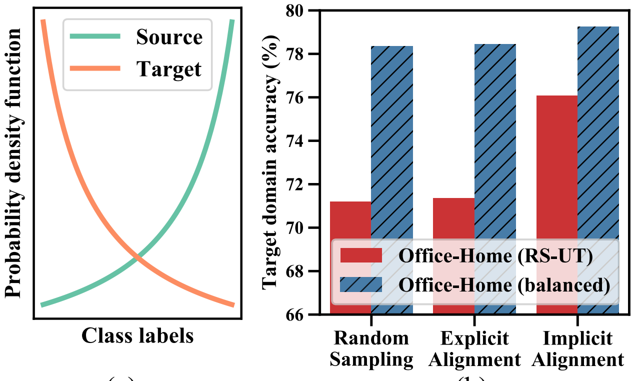
Baselines. Our main explicit alignment baselines are COAL [Tan et al., 2019], PACET [Liang et al., 2019b] and MCS [Liang et al., 2019a], state-of-the-art explicit alignment methods based on domain discriminator discrepancy. For the class distribution shift experiments, our main baseline is COAL [Tan et al., 2019], an explicit alignment method designed for class class distribution shift. For the experiments with the standard datasets, our main baselines are PACET [Liang et al., 2019b] and MCS [Liang et al., 2019a], state-of-the-art explicit alignment algorithms based on domain discriminator-based discrepancy measures. As our domain discrepancy measure is MDD, we re-implement various MDD-based explicit alignment for fair comparison.

Computational efficiency. Self-training requires the estimation of target domain labels, which could be time-consuming depending on the size of the target domain. To improve the computational efficiency of our algorithm, we only update pseudo-labels periodically, i.e., every 20 steps, instead of at every training step. We show in Section 3.5.4 that our method does not require more frequent pseudo-label updates as they exhibit similar performance on the target domain. We leave the caching and updating strategies of pseudo-labels as future work.

3.3.2 Results and Discussions

Extreme Class Distribution Shift

We use Office-Home (RS-UT), described in Figure 3.3 (a), to evaluate the performance of different methods under extreme within-domain class imbalance and between-domain class distribution shift where the majority classes in the source domain are minority classes in the target domain. Table 3.1 presents the per-class average accuracy on Office-Home (RS-UT). Our main baseline is the explicit alignment method "covariate and label shift co-alignment" (COAL) designed to address class distribution shift. Our proposed implicit domain alignment consistently outperforms previous state-of-the-art approaches.



(a) (b)

Figure 3.3: (a) Source and target class distribution of Office-Home (RS-UT). (b) Accuracy comparison between Office-Home (RS-UT) and Office-Home (balanced) for Rw→Pr.

Table 3.1: Per-class average accuracy on Office-Home dataset with RS-UT label shifts (ResNet-50). Due to the computational complexity of the experiments, the reported results are from a single random seed. The performance is not sensitive to different random seeds.

Methods	**Rw→Pr**	**Rw→Cl**	**Pr→Rw**	**Pr→Cl**	**Cl→Rw**	**Cl→Pr**	Avg
Source Only†	69.77	38.35	67.31	35.84	53.31	52.27	52.81
BSP [Chen et al., 2019c]†	72.80	23.82	66.19	20.05	32.59	30.36	40.97
PADA [Cao et al., 2018]†	60.77	32.28	57.09	26.76	40.71	38.34	42.66
BBSE [Lipton et al., 2018]†	61.10	33.27	62.66	31.15	39.70	38.08	44.33
MCD [Saito et al., 2018]†	66.03	33.17	62.95	29.99	44.47	39.01	45.94
DAN [Long et al., 2015]†	69.35	40.84	66.93	34.66	53.55	52.09	52.90
F-DANN [Wu et al., 2019]†	68.56	40.57	67.32	37.33	55.84	53.67	53.88
JAN [Long et al., 2017]†	67.20	43.60	68.87	39.21	57.98	48.57	54.24
DANN [Ganin et al., 2016]†	71.62	46.51	68.40	38.07	58.83	58.05	56.91
MDD (random sampler)	71.21	44.78	69.31	42.56	52.10	52.70	55.44
MDD (source-balanced sampler)	76.06	47.38	71.56	40.03	57.46	58.54	58.50
COAL [Tan et al., 2019]†,‡	73.65	42.58	73.26	40.61	59.22	57.33	58.40
MDD+Explicit Alignment (basic)‡	69.52	44.70	69.59	40.27	53.02	53.39	55.08
MDD+Explicit Alignment (moving avg.)‡	71.37	45.26	69.69	40.28	52.92	52.69	55.37
MDD+Explicit Alignment (curriculum)‡	70.02	45.48	69.71	40.86	53.26	52.99	55.39
MDD+Implicit Alignment	**76.08**	**50.04**	**74.21**	**45.38**	**61.15**	**63.15**	**61.67**

† *Source*: Data of these baseline methods are cited from [Tan et al., 2019].
‡ Methods using explicit class-conditioned domain alignment.

The impact of class distribution shift

Many baseline methods suffer from class distribution shift, and their performances degrade to "Source Only" training because they do not take into account between-domain class distribution shift. For MDD-based methods, after we apply balanced sampling for the source domain, the per-class average accuracy improved from 55.44% to 58.50%, which indicates balanced sampling is helpful for class distribution shift, despite only in the source domain.

The effectiveness of implicit alignment

The effectiveness of implicit alignment is demonstrated through the comparison between "MDD+Implicit Alignment" with "MDD (source-balanced sampler)". Both methods use the same sampling procedure for the source domain. Their only difference is that implicit alignment aligns the two domains by selectively sampling aligned classes in the target domain, whereas "source-balanced sampler" only takes random samples from the target domain. Table 3.1 shows that implicit alignment performs better than "source-balanced sampler" because it is better-aligned. Besides, the proposed method also outperforms MDD-based explicit alignment baselines, which further validates the effectiveness of implicit alignment over the explicit alignment.

Robustness to class distribution shift

Figure 3.3 (b) compares the baseline, implicit and explicit alignments on Office-Home (balanced) and Office-Home (RS-UT). We observe that implicit alignment performs the best on both datasets. More importantly, implicit alignment is more robust to class distribution shift which greatly out-performs other methods under RS-UT distribution shift and has a smaller performance drop from the balanced version of Office-Home.

Evaluation on Standard datasets

Table 3.2 and Table 3.3 summarize the results for the standard Office-31 and Office-Home datasets which have a small degree of class imbalance. Our method outperforms the baselines in 3 out of 6 domain pairs for Office-31, and 10 out of 12 domain pairs for Office-Home (standard). The proposed implicit alignment exhibits larger performance gains on the Office-Home dataset because the dataset is more difficult for domain adaptation, and it has 65 classes compared with the 31 classes in Office-31. Our method is especially useful for tasks with a large number of classes because not all classes can fit in one batch, which necessitates sampling-based alignment for better training. We show, in the supplementary materials, that our method not only converges better but also converges faster than the baseline methods as well.

Table 3.2: Accuracy with 95% confidence interval (%) on Office-31 (standard) for unsupervised domain adaptation (ResNet-50). We repeated each experiment 5 times with different random seeds and report the average and the standard error of the accuracy. Numbers in **bold** represent statistical significant results.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
Source only	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN [Long et al., 2015]	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
DANN [Ganin et al., 2016]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA [Tzeng et al., 2017]	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN [Long et al., 2017]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MADA [Pei et al., 2018]	90.0 ± 0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
GTA [Sankaranarayanan et al., 2018]	89.5±0.5	97.9±0.3	99.8±0.4	87.7±0.5	72.8±0.3	71.4±0.4	86.5
MCD [Saito et al., 2018]	88.6±0.2	98.5±0.1	**100.0**±.0	92.2±0.2	69.5±0.1	69.7±0.3	86.5
CDAN [Long et al., 2018]	94.1±0.1	98.6±0.1	**100.0**±.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
MDD [Zhang et al., 2019b]	**94.5**±0.3	98.4±0.1	**100.0**±.0	**93.5**±0.2	74.6±0.3	72.2±0.1	**88.9**
PACET [Liang et al., 2019b]‡	90.8	97.6	99.8	90.8	73.5	73.6	87.4
CAT [Deng et al., 2019]‡	94.4±0.1	98.0±0.2	**100.0**±0.0	90.8±1.8	72.2±0.2	70.2±0.1	87.6
MDD (source-balanced sampler)	90.4±0.4	**98.7**±0.1	99.9±0.1	90.4±0.2	75.0±0.5	73.7±0.9	88.0
MDD+Explicit Alignment‡	92.3±0.1	98.2±0.1	99.8±.0	92.3±0.3	74.6±0.2	72.9±0.7	88.4
MDD+Implicit Alignment	90.3±0.2	**98.7**±0.1	99.8±.0	92.1±0.5	**75.3**±0.2	**74.9**±0.3	88.8

‡ Methods using explicit class-conditioned domain alignment.

Table 3.3: Accuracy (%) on Office-Home (standard) for unsupervised domain adaptation (ResNet-50). Due to the computational complexity of the experiments, the reported results are from a single random seed. The performance is not sensitive to different random seeds.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
Source only	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [Long et al., 2015]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [Ganin et al., 2016]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [Long et al., 2017]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [Long et al., 2018]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BSP [Chen et al., 2019c]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
MDD [Zhang et al., 2019b]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	**72.5**	**60.2**	82.3	68.1
MCS [Liang et al., 2019a]‡	55.9	73.8	79.0	57.5	69.9	71.3	58.4	50.3	78.2	65.9	53.2	82.2	66.3
MDD+Explicit Alignment‡	54.3	74.6	77.6	60.7	71.9	71.4	62.1	52.4	76.9	71.1	57.6	81.3	67.7
MDD (source-balanced sampler)	55.3	75.0	79.1	62.3	70.1	73.2	63.5	53.2	78.7	70.4	56.2	82.0	68.3
MDD+Implicit Alignment	**56.2**	**77.9**	**79.2**	**64.4**	**73.1**	**74.4**	**64.2**	**54.2**	**79.9**	71.2	58.1	**83.1**	**69.5**

‡ Methods using explicit class-conditioned domain alignment.

The impact of source-balanced sampling

Similar to findings in extreme class distribution shift (Section 3.3.2), we observe source-balanced sampling is helpful when comparing "MDD (source-balanced sampler)" with the MDD standard baseline, even without extreme class distribution shift.

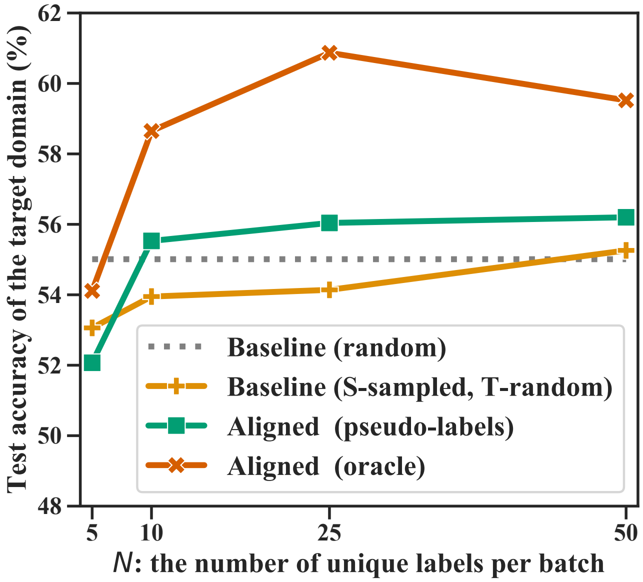


Figure 3.4: The impact of class diversity and alignment on domain adaptation for Ar→Cl, Office-Home (standard). Due to the computational complexity of the experiments, the reported results are from a single random seed. The performance is not sensitive to different random seeds.

Comparison with explicit alignment

The proposed method outperforms the state-of-the-art explicit alignment methods—i.e., PACET and MCS—across all domain pairs. We find it ineffective to incorporate prototype-based explicit alignment into MDD. This is in contrast with domain-discriminator-based adversarial learning, where explicit alignment is shown to improve domain adaptation. We argue this is because the classifier-based discrepancy MDD contains more abundant information than domain-discriminator-based discrepancy, owing to the availability of predictive probabilities provided by the classifiers. The rich information in domain discrepancy removes the need to augment the adversarial loss with prototype-based distances in the Euclidean space. Moreover, we find that explicit alignment is very sensitive to the weight λ of the alignment loss. We experimented with different values of $\lambda \in [0.1, 0.01, 0.001, 0.0001, 0.00001]$ and chose 0.0001 as the best performing one for "MDD+Explicit Alignment".

Impact of class diversity and alignment

We analyze the impact of class diversity and alignment by designing experiments along three dimensions: the number of unique labels in each minibatch, whether the classes are aligned, and whether we use pseudo-labels or ground-truth labels when sampling the target domain. Note that all experiments in Figure 3.4 have the same batch size 50. **Setup.** "Baseline (random)" randomly samples examples of both domains. "Baseline (S-sampled, T-random)" uses N-way sampler for the source domain, and randomly samples the target domain. "Aligned (pseudo-labels)" is the proposed implicit alignment approach. "Aligned (Oracle)" is the oracle form of implicit alignment where the target domain uses ground-truth labels for sampling.

The impact of class diversity. Minibatch-based class diversity determines the sampling distribution of the label space, and a greater diversity corresponds to a more stable measure of this sampling distribution. Figure 3.4 suggests a positive correlation between the model performance and class diversity: domain adaptation methods do not work well when the class diversity is very low—i.e., only sample 5 classes per batch among the 65 classes—and the alignment-based methods outperform the baseline as we increase class diversity.

The impact of alignment. Quantitative Results in Figure 3.4 confirms the importance of the proposed implicit alignment algorithm from two perspectives. First, "Aligned (oracle)" consistently performs the best, which suggests perfect alignment can provide substantial benefits to unsupervised domain adaptation. Second, the comparison between "Aligned (pseudo-labels)" and "Baseline (S-sampled, T-random)" validates the effectiveness of pseudo-label based implicit alignment, although the pseudo-labels are approximations of the oracle. As noted in previous experiments, aligning the source and target domains is beneficial to domain adaptation.

Robustness to pseudo-label errors

We investigate whether implicit alignment is indeed more robust to pseudo-label errors when compared with explicit alignment. Figure 3.5 illustrates the relationship between pseudo-label accuracy at training step t and the corresponding subsequent target accuracy at step $t+1000$, i.e., after 1000 domain adaptation training steps. This process resembles a Markov chain that allows us to analyze the impact of pseudo-label accuracy on the learning dynamics.

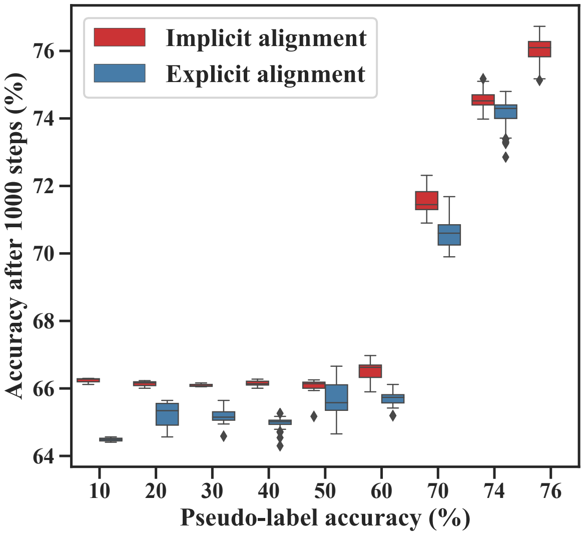


Figure 3.5: The impact of pseudo-label errors on implicit and explicit alignment, Office-Home (standard).

It is evident from Figure 3.5 that the drawbacks of explicit alignment are more severe when the pseudo-labels are less accurate, e.g., 10∼40% on the x-aixs, where implicit alignment has more considerable performance improvements than explicit alignment. This suggests that implicit alignment is more robust to erroneous pseudo-label predictions because it does not require explicit supervision from the pseudo-labels. Implicit and explicit methods eventually converge at 76% and 74%, respectively. The results are significant at the 95% confident interval with different random seeds. Note that the missing box for “Explicit Alignment” at 76% is because “Explicit Alignment” never reaches 76% pseudo-label accuracy. Because the use of pseudo-labels inevitably introduces bias to the learning system, to further reduce the bias, one could use an unsupervised approach such as conditional variational autoencoder to generate more accurate class-conditioned samples.

Although many recent techniques attempt to address pseudo-label bias in explicit alignment, they depend on the assumption that probabilities of model predictions are well-calibrated during training. They fail to address ill-calibrated probabilities [Guo et al., 2017], where the model tends to make confident mistakes on the target domain. Moreover, given that models do not initially perform well when training begins, for a random classifier, implicit alignment selects random samples that is equivalent to training without sampling. In contrast, explicit alignment optimizes model parameters from these random labels explicitly.

	Alignment options		
Domains	masking	sampling	Accuracy
Rw→Cl	×	×	44.8
	√	×	44.8
	×	√	47.4
	√	√	**50.0**
Pr→Rw	×	×	69.3
	√	×	72.7
	×	√	72.0
	√	√	**74.2**

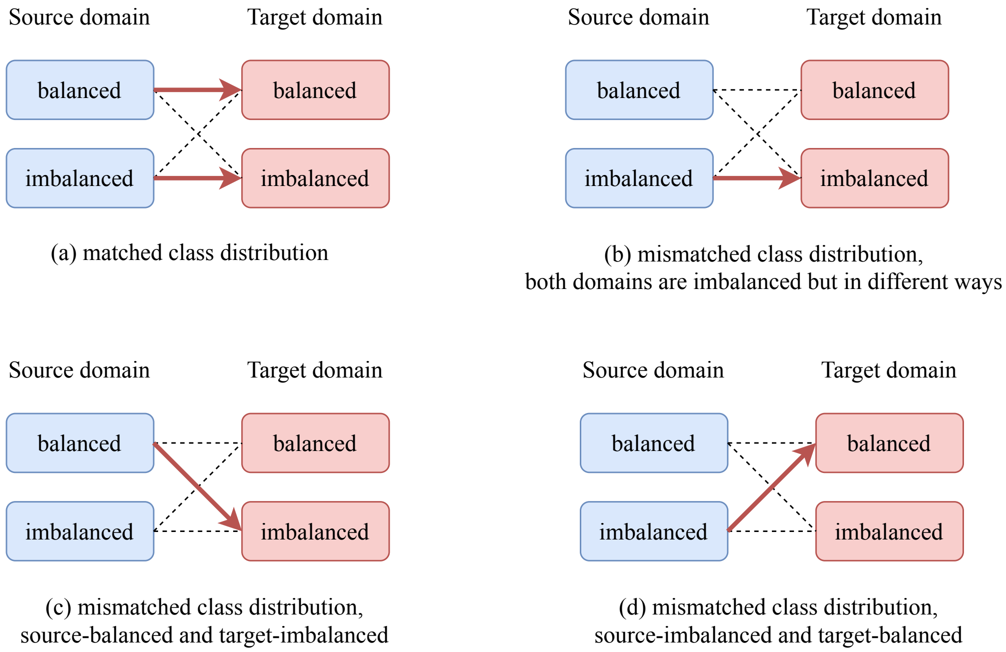
Table 3.4: The impact of different implicit alignment options, i.e., masking in the MDD estimation and sampling class-aligned minibatches, on Office-Home (RS-UT). Evaluated on average accuracy per-class.

Ablation Study on Implicit Alignment

Table 3.4 presents the ablation study on Office-Home (RS-UT) that aims to assess the impact of different implicit alignment options: alignment in the domain divergence estimations in Section 3.2.3 (i.e., *masking* in MDD) and alignment in the input space in Section 3.2.3 (i.e., *sampling* class-conditioned examples). We observe that both alignment techniques are essential for domain adaptation because alignment should be enforced consistently across all aspects of the domain adaptation training. We report similar findings in the supplementary materials for the standard Office-Home dataset.

3.4 Synthetic Experiments on MNIST and SVHN

We design extensive experiments to further demonstrate the effectiveness of the proposed method. The class distributions of SVHN [Netzer et al., 2011] and MNIST [LeCun et al., 2010] are synthetically manipulated to simulate various interactions between *within-domain class imbalance* and *between-domain class distribution shift*. The four scenarios are depicted in Figure 3.6. We choose the domain pair MNIST and SVHN as they are the most challenging domain pairs in the digits domain adaptation dataset.



(a) matched class distribution

(b) mismatched class distribution,
both domains are imbalanced but in different ways

(c) mismatched class distribution,
source-balanced and target-imbalanced

(d) mismatched class distribution,
source-imbalanced and target-balanced

Figure 3.6: Interactions between within-domain class imbalance and between-domain class distribution shift.

3.4.1 Setup

We use DANN as the adversarial training algorithm instead of MDD for generality. When the source and target domains have match priors where $p_S(y)=p_T(y)$, the class distributions can be either balanced or imbalanced. For mismatched class distribution shift where $p_S(y)\neq p_T(y)$, we simulate three types of between-domain distribution shift:

- source-balanced, target-imbalanced;
- source-imbalanced, target-balanced;
- both-imbalanced.

We also simulate different degrees of within-domain class imbalance:

- light: light-tailed class imbalance from a triangular-like distribution;
- heavy: heavy-tailed class imbalance from a Pareto distribution.

3.4.2 Results on Matched Class Distribution

Table 3.5 presents the per-class accuracy of the target domain when $p_S(y)=p_T(y)$. The proposed "DANN+implicit" is has similar performance with the baseline "DANN" on the doomain pair SVHN→MNIST, but outperforms the "DANN" on the SVHN→MNIST domain pair. It is worthwhile to note that SVHN→MNIST is a particularly difficult domain pair where domain adaptation methods, including DANN, tend to perform worse than the "source only" baseline. In other words, conventional domain adaptation methods are detrimental to the domain pair SVHN→MNIST while the proposed implicit alignment overcomes this limitation and greatly improves the performance of DANN.

Table 3.5: Per-class average accuracy (%) with *matched prior* where $p_S(y)\!=\!p_T(y)$. Numbers in **bold** represent statistical significant results.

	SVHN→MNIST			MNIST→SVHN		
method	balanced	light imbalance	heavy imbalance	balanced	light imbalance	heavy imbalance
source only	72.0±2.5	57.3±3.5	47.6±8.3	**31.4**±1.6	31.2±0.6	**28.8**±0.6
DANN	**89.3**±1.2	**87.5**±0.9	**69.1**±4.4	23.2±1.0	22.0±2.9	22.0±2.7
DANN+implicit	**92.3**±2.1	**87.3**±1.3	**66.8**±6.9	**33.2**±4.7	**36.3**±1.8	**33.9**±6.1

3.4.3 Results on Mismatched Class Distribution

Table 3.6, 3.7 and 3.8 present the per-class average classification accuracy with 95% confidence interval (%) for mismatched class distributions in Figure 3.6 (b) (C), and (d), respectively. Numbers in **bold** represent statistical significant results. We find that the proposed method significantly improves (10-30%) the domain adaptation performance of DANN regardless of the degree of class imbalance or the type of distribution shift. We also find that implicit alignment provides greater improvements as the degree of class imbalance becomes more severe, i.e., comparison between "light imbalance" and "heavy imbalance". Furthermore, similar to findings in Section 3.4.2, implicit alignment overcomes this limitation of DANN greatly improves the performance for the challenging task SVHN→MNIST.

Table 3.6: Per-class average accuracy (%) with *mismatched prior* where $p_S(y)\!\neq\!p_T(y)$ and both domains are imbalanced. Numbers in **bold** represent statistical significant results.

	SVHN→MNIST		MNIST→SVHN	
method	light imbalance	heavy imbalance	light imbalance	heavy imbalance
source only	60.9±5.2	51.2±5.9	30.6±1.3	**27.1**±1.7
DANN	67.6±0.8	40.5±5.5	23.4±1.6	18.8±2.9
DANN+implicit	**88.6**±0.6	**70.5**±3.6	**36.3**±2.5	**27.9**±2.4

Table 3.7: Per-class average accuracy (%) with *mismatched prior* where $p_S(y) \neq p_T(y)$. The source domain is balanced while the target domain is imbalanced. Numbers in **bold** represent statistical significant results.

	SVHN→MNIST		MNIST→SVHN	
method	light imbalance	heavy imbalance	light imbalance	heavy imbalance
source only	67.4±7.3	66.3±3.3	**32.5**±2.9	**28.2**±2.3
DANN	78.2±2.8	59.1±0.8	20.9±6.0	20.5±3.1
DANN+implicit	**88.6**±0.7	**82.2**±2.1	**32.4**±2.1	**28.9**±3.3

Table 3.8: Per-class average accuracy (%) with *mismatched prior* where $p_S(y) \neq p_T(y)$. The source domain is imbalanced while the target domain is balanced. Numbers in **bold** represent statistical significant results.

	SVHN→MNIST		MNIST→SVHN	
method	light imbalance	heavy imbalance	light imbalance	heavy imbalance
source only	65.2±2.1	53.3±1.3	**31.6**±3.3	**32.8**±0.9
DANN	82.0±0.7	52.3±2.3	23.4±3.6	25.9±0.5
DANN+implicit	**91.0**±1.9	**87.1**±2.6	**34.9**±0.5	**31.1**±2.9

3.5 Supplementary Empirical Results

3.5.1 Additional Evaluation Measures on Office-Home

Table 3.9: Additional evaluation measures on Office-Home (%) with ResNet-50.

	Ar→Cl		Pr→Rw	
	MDD	ours	MDD	ours
accuracy	54.91	56.17	77.46	79.94
macro F1 score	53.66	55.29	75.86	78.42
weighted F1 score	53.97	55.81	77.24	79.79
macro precision	57.02	57.72	78.21	79.56
weighted precision	58.85	60.30	79.60	80.97
macro recall	56.41	57.76	76.30	78.61
weighted recall	54.91	56.17	77.65	79.94

Table 3.9 presents additional evaluation on Office-Home (standard). We re-implement MDD using identical batch sizes (50) and random seeds for fair comparison. The results

show that our proposed method has consistent improvements across all evaluation measures, and the improvements are not a result of batch sizes or random seeds.

3.5.2 Ablation on Alignment Options

Table 3.10: The impact of different implicit alignment options, i.e., masking the classifier-based domain discrepancy measure and sampling examples from the source and target domains, on Ar→Cl and Cl→Pr, Office-Home (standard).

	Alignment options		
Domains	masking	sampling	Accuracy
Ar→Cl	×	×	55.3
	√	×	55.5
	×	√	54.6
	√	√	**56.2**
Cl→Pr	×	×	71.4
	√	×	70.1
	×	√	70.5
	√	√	**73.1**

Table 3.10 presents the ablation study on Office-Home (standard) that aims to assess the impact of different implicit alignment options: alignment in the domain divergence estimations (i.e., *masking* in MDD) and alignment in the input space (i.e., *sampling* class-conditioned examples). We observe that both alignment techniques are essential for domain adaptation because alignment should be enforced consistently across all aspects of the domain adaptation training. This is consistent to findings in the main chapter.

3.5.3 Learning Curve

Figure 3.7 shows the learning curve of the target domain accuracy for different methods. We find that explicit alignment has similar performances with the baseline, while implicit alignment performs the best.

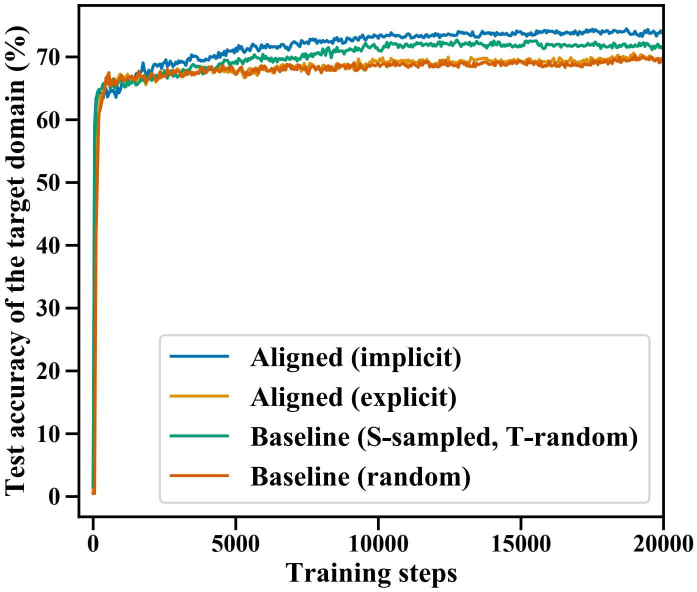


Figure 3.7: Learning curve of the target domain accuracy for Pr→Rw, Office-Home (RS-UT).

Table 3.11: The impact of pseudo-label update frequency on Ar→Cl, Office-Home (standard).

pseudo-labels updated every N steps	accuracy
5	56.0
10	56.7
20	56.2
50	55.2
100	56.3
500	55.7

3.5.4 Computational Efficiency

Self-training requires estimating the target domain labels, which could be time-consuming depending on the size of the dataset. To improve the computational efficiency of our algorithm, we only update pseudo-labels periodically, i.e., every 20 steps, instead of at every training step. We show in Table 3.11 that different pseudo-label update frequencies exhibit similar performance on the target domain. Notably, implicit alignment outperforms the baseline method in spite of only updating the pseudo-labels every 500 training steps. This validates the robustness of implicit alignment.

For the experiments described in Section 3.5.3, training the baseline methods take

31 hours (wall clock time), whereas implicit alignment takes 34 hours under the same training condition when the pseudo-labels are updated every 20 steps. The 10% computational overhead is rather restricted. Moreover, from an engineering perspective, partially updating and caching the pseudo-labels could further improve the computational efficiency, and we leave them as future work. Sweep rehearsal [Robins, 1995; Silver et al., 2015] is a promising direction that keeps a small dynamic buffer of examples to improve the efficiency of pseudo-labeled examples.

3.5.5 The Impact of Batch Size

Table 3.12: The impact of batch size on target domain accuracy (%), Ar→Cl, Office-Home (standard). The MDD results are based on our re-implementation.

batch size	baseline	implicit
8	48.9	49.7
16	52.7	52.8
32	54.9	56.2
50	55.3	56.2

Table 3.12 presents the impact of batch size on the target domain accuracy. We find that implicit alignment consistently improves the model performance over the MDD baseline across different batch sizes, and both methods work better with larger batch sizes. This is consistent with our findings on empirical class diversity where a greater diversity corresponds to a more stable measure of this sampling distribution, which in turn leads to better domain adaptation performances.

There is a positive correlation between the batch size and model performance, which is consistent with the probably approximately correct (PAC) framework [Valiant, 1984]. Due to the GPU's memory constraints, we were not able to experiment with larger batch sizes on this dataset to verify whether the two approaches would converge to the same performance. It is worth noting that the proposed approach is shown to improve the domain adaptation performance when evaluated on various datasets where each dataset uses a different batch size—especially for the digits experiments in Table 3.6, 3.7 and 3.8 where the batch size is relatively large, for example, 100.

3.5.6 Empirical Class Diversity

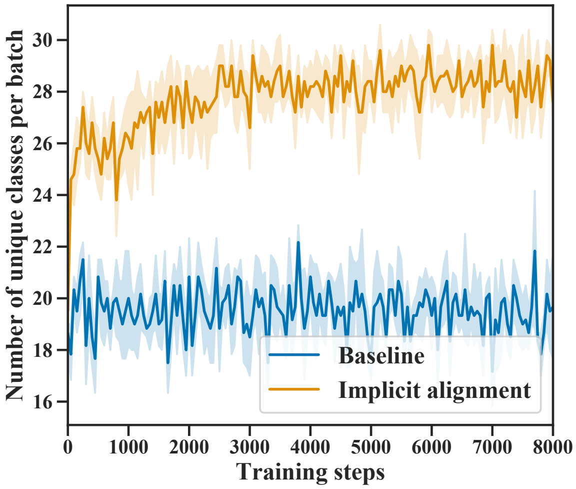


Figure 3.8: Empirical class diversity while training A→W (Office-31) with batch size 31.

Figure 3.8 shows the empirical class diversity comparing implicit alignment with the baseline. In both experiments, the batch size is identical with the total number of classes (i.e., 31). For the baseline method, random sampling only obtains about 19 unique classes per-batch, which is much smaller than the batch size, in spite of the batch sizes being the same with the total number of classes. This is because random sampling can be viewed as sampling *with replacement* in the label space, whereas the implicit alignment can be viewed as sampling *without replacement* in the label space, which naturally increases the empirical class diversity. The expected class diversity of the baseline is

$$\mathbb{E}[|Y|] = n\left[1 - \left(\frac{n-1}{n}\right)^k\right], \tag{3.17}$$

where n is the number of unique classes and k is the size of the minibatch. The expected class diversity is 19.78 if $n=31$ and $k=31$, which is consistent with the empirical class diversity shown in Figure 3.8.

For the implicit alignment method shown in Figure 3.8, although it has low class diversity at training step 0 due to the random pseudo-labels, it has a sharp increase in class diversity for the first few hundred training steps, and eventually being able to sample 28 classes from the total of 31 classes. This confirms that implicit alignment is effective in improving empirical class diversity beyond random sampling.

3.5.7 VisDA2017 Dataset

We also report the results for VisDA [Peng et al., 2017], a synthetic to real domain adaptation task with 12 classes, in Table 3.13. The proposed implicit alignment has a 4.7% improvement over our re-implementation of the MDD baseline. Note that we did not perform any hyper-parameter tuning for this dataset and we use the same hyper-parameters with our Office-Home experiments.

Table 3.13: Target domain accuracy (%) on VisDA2017 (synthetic→real)

method	accuracy
JAN [Long et al., 2017]	61.6
GTA[Sankaranarayanan et al., 2018]	69.5
MCD [Saito et al., 2018]	69.8
CDAN [Long et al., 2018]	70.0
MDD [Zhang et al., 2019b]	74.6
MDD (our re-implementation)	65.0
MDD+implicit	69.7

3.6 Limitations

Although the proposed approach can provide class-conditioned domain alignment between the source and target domains, it is not able to deal with intra-class subtype distribution mismatch where the distribution of subtypes differs between the source and target domains. This problem is also known as “hidden stratification” and has been shown to “Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging” [Oakden-Rayner et al., 2020]. As an example, for X-ray images classification tasks, if we only have access to “healthy” and “unhealthy” labels, but do not have access to the underlying distribution of different subtypes of the “unhealthy” category, the adaptation might cause risks for some subclasses. More work needs to be done in unsupervised domain adaptation to ensure its reliability in safety critical applications.

3.7 Conclusion

We introduce an approach for unsupervised domain adaptation—with a strong focus on practical considerations of between-domain class distribution shift—from a class-conditioned domain alignment perspective. We show theoretically that the proposed implicit alignment provides a more reliable measure of empirical domain divergence which facilitates adversarial domain-invariant representation learning, that would otherwise be hampered by the class-misaligned domain divergence. We extend our theory on implicit alignment into the classifier-based domain divergence measure and provide extensive experiments to show that our proposed approach leads to superior UDA performance under extreme within-domain class imbalance and between-domain class distribution shift, as well as competitive results on standard UDA tasks. We emphasize that the proposed method is robust to pseudo-label bias, simple to implement, has a unified training objective, and does not require additional parameter tuning.

Chapter 4

Learning to Learn with Conditional Class Dependencies

4.1 Introduction

In machine learning, the objective of classification is to train a model to categorize inputs into various classes. We usually assume a categorical distribution over the label space, and thus effectively ignore dependencies among them. However, class structure does exist in real world and is also present in most datasets. Although class structure can be implicitly obtained as a by-product during learning, it is not commonly exploited in an explicit manner to develop better learning systems. The use of label structure might not be of prime importance when having access to huge amounts of data, such the full ImageNet dataset. However, in the case of few-shot learning where little data is available, meta-information such as dependencies in the label space can be crucial.

In recent years, few-shot learning—learning from few examples across many tasks—has received considerable attention [Finn et al., 2017a; Ravi and Larochelle, 2016; Snell et al., 2017b; Vinyals et al., 2016]. In particular, the concept of meta-learning has been shown to provide effective tools for few-shot learning tasks. In contrast to common transfer learning methods that aim to fine-tune a pre-trained model, meta-learning systems are trained by being exposed to a large number of tasks and evaluated in their ability to learn new tasks effectively. In meta-training, learning happens at two levels: a meta-learner that learns across many tasks, and a base-learner that optimizes for each task. Model-Agnostic Meta-Learning (MAML) is a gradient-based meta-learning algorithm that provides a mechanism for rapid adaptation by optimizing only for the initial parameters of the base-learner [Finn et al., 2017a].

Our motivation stems from a core challenge in gradient-based meta-learning, wherein the quality of gradient information is key to fast generalization: it is known that gradient-based optimization fails to converge adequately when trained from only a few examples [Ravi and Larochelle, 2016], hampering the effectiveness of gradient-based meta-learning techniques. We hypothesize that under such circumstances, introducing a metric space trained to encode regularities of the label structure can impose global class dependencies on the model. This class structure can then provide a high-level view of the input examples, in turn leading to learning more disentangled representations.

We propose a meta-learning framework taking advantage of this class structure information, which is available in a number of applications. The Conditional class-Aware Meta-Learning (CAML) model is tasked with producing activations in a manner similar to a standard neural network, but with the additional flexibility to shift and scale those activations conditioned on some auxiliary meta-information. While there are no restrictions on the nature of the conditioning factor, in this work we model class dependencies by means of a metric space. We aim to learn a function mapping inputs to a metric space where semantic distances between instances follow an Euclidean geometry—classes that are semantically close lie in close proximity in an ℓ^p sense. The goal of the conditional class-aware transformation is to make explicit use of the label structure to inform the model to reshape the representation landscape in a manner that incorporates a global sense of class structure.

The contributions of this work are threefold: (i) We provide a meta-learning framework that makes use of structured class information in the form of a metric space to modulate representations in few-shot learning tasks; (ii) We introduce class-aware grouping to improve the statistical strength of few-shot learning tasks; (iii) We show experimentally that our proposed algorithm learns more disentangled representation and achieves competitive results on the *mini*ImageNet benchmark.

4.2 Method

As shown in Figure 4.1, the proposed Conditional class-Aware Meta-Learning (CAML) is composed of four components: an embedding function f_ϕ that maps inputs to a metric space, a base-learner f_θ that learns each individual task, an adaptation function f_c that conditionally modulates the representations of the base-learner, and a meta-learner that learns across different tasks. Figure 4.1 depicts a toy illustration of the task inference procedure where examples from three classes are mapped onto a metric space using f_ϕ, which are further used to modulate the base-learner f_θ through a conditional transformation function f_c.

The main contribution of this chapter is to incorporate metric-based conditional

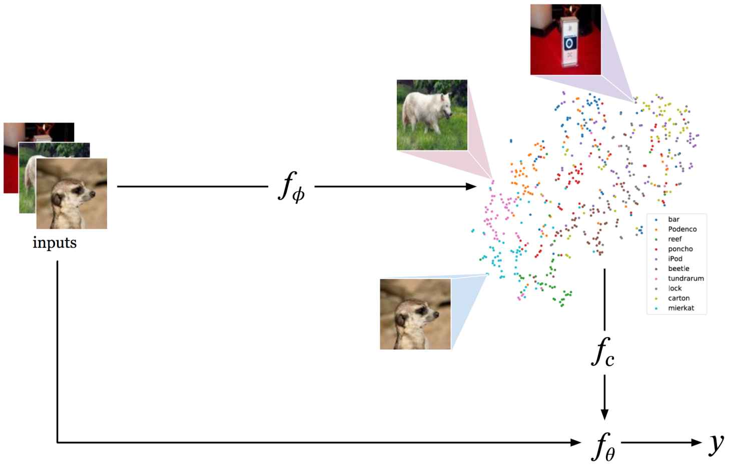


Figure 4.1: Overview of Conditional class-Aware Meta-Learning. Inputs to the model are mapped onto an embedding space using f_ϕ which are then used to modulate the base-learner f_θ through a conditional transformation f_c. We use MAML (not shown) to meta-learn f_c, f_θ, and a metric loss to pretrain f_ϕ

transformations (f_c) into the meta-learning framework at the instance level. A notable feature of the proposed method is that the model has a global sense of the label space through the embedding function f_ϕ by mapping examples onto the semantically meaningful metric space. The embeddings on the metric space inform the base-learner f_θ about the label structure which in turn helps disentangle representations from different classes. This structured information can also provide a global view of the input examples to improve gradient-based meta-learning.

In a simplistic form, our proposed model makes predictions using

$$\hat{y} = f_\theta\Big(x; f_c\big(f_\phi(x)\big)\Big),$$

where the base-learner f_θ is conditioned on the embedding space $f_\phi(x)$ through the conditional transformation f_c. This is in contrast with a regular base-learner where $\hat{y} = f_\theta(x)$. In our framework, we use MAML to meta-learn f_c and f_θ. The metric space is pretrained using distance-based loss function. The setup is similar to multitask learning in that all model parameters are learned jointly. They differ in that, in multitask learning, the model parameters are shared across tasks. In contrast, in meta-learning, the meta-parameters are shared through initialization, but each model has its own set of task-specific parameters.

4.2.1 Metric Space as Conditional Information

We encode information of the label structure through f_ϕ in the form of an M-dimensional metric space, where each input example is reduced to a point in the metric space. To learn a semantically meaningful metric space, we first define a centroid $\mathbf{c}_t$ for each class t,

$$\mathbf{c}_t = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}^{\text{train}}} \mathbb{1}_{\{y_i = t\}} f_\phi(\mathbf{x}_i),$$

where K denotes the number of examples for class t, $\mathbb{1}_{\{y_i=t\}}$ denotes an indicator function of y_i which takes value 1 when $y_i = t$ and 0 otherwise. The mapping function f_ϕ is optimized to minimize the negative log-probability defined in Eq. (4.1) by minimizing the Euclidean distance d between an example and its corresponding class centroid $\mathbf{c}_t$ while maximizing its Euclidean distance to other class centroids $\mathbf{c}_{t'}$:

$$\underset{\phi}{\operatorname{argmin}} \mathbb{E}\left[d(f_\phi(\mathbf{x}_i), \mathbf{c}_t)) + \log \sum_{t'} \exp(-d(f_\phi(\mathbf{x}_i), \mathbf{c}_{t'})) \right]. \tag{4.1}$$

In relation to prototypical networks [Snell et al., 2017b], we use the same loss function for metric learning. However, these frameworks differ in the test mode: we are not interested in example-centroid distances for label assignment, but rather in the projection $f_\phi(\mathbf{x}_i)$ from the input space to the metric space that encapsulates inferred class regularities given the input example $\mathbf{x}_i$.

4.2.2 Conditionally Transformed Convolutional Block

Conditional transformations have previously been explored in style transfer and visual reasoning. Instead of learning different styles with separate networks, conditional instance normalization allows different styles to share the same conditional style transfer network so that different styles are characterized by different scale and shift parameters of the feature maps. In visual question answering, De Vries et al. [2017] have shown that it is beneficial to modulate early visual signals of a pre-trained residual network by language in the form of conditional batch normalization. It was further shown that feature-wise linear modulation [Dumoulin et al., 2018; Perez et al., 2017] can efficiently select meaningful representations for visual reasoning.

The notion that is common to all these methods is the use of an additional input source, e.g., style or language, to conditionally transform intermediate representations

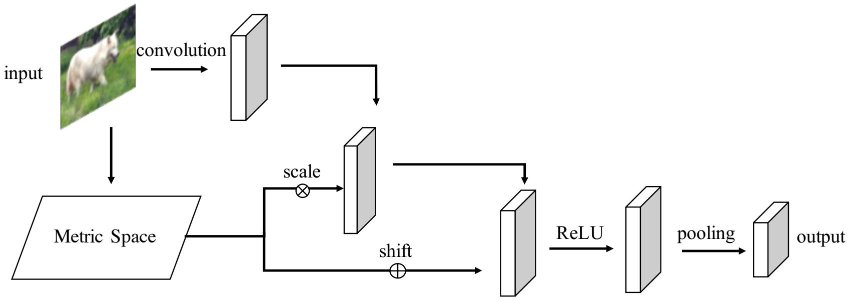


Figure 4.2: Conditionally transformed convolutional block. The convolutional feature maps are conditionally scaled and shifted based on the input image's representation in the metric space.

of a network. In few-shot learning, Zhou et al. [2018] suggested that it is easier to operate in the concept space in the form of a lower dimensional representation. This is compatible with our proposed approach that uses the metric space as concept-level representation to modulate intermediate features of the base-learner.

We now turn to describing the conditionally transformed convolutional block, shown in Figure 4.2, which uses the metric space described in Section 4.2.1 to inform the base-learner about the label structure of a task. The conditional transformation f_c receives embeddings from the metric space and produces transformation operations to modulate convolutional representations of the base-learner f_θ.

Our conditional transformation has close relation to Batch Normalization (BN) [Ioffe and Szegedy, 2015] that normalizes the input to every layer of a neural network. In order to conditionally modulate feature representations, we use Conditional Batch Normalization (CBN) [Dumoulin et al., 2017] to predict scale and shift operators from conditional input $\mathbf{s}_i$:

$$\hat{\gamma}_c = f_{c,\gamma}(\mathbf{s}_i), \qquad \hat{\beta}_c = f_{c,\beta}(\mathbf{s}_i), \tag{4.2}$$

where $f_{c,\gamma}$ and $f_{c,\beta}$ can be any differentiable function. This gives our model the flexibility to shift or scale the intermediate representations based on some source information in $\mathbf{s}_i$. Since examples belonging to the same class are conceptually close, we exploit this inherent relationship in the metric space to modulate the feature maps at the example level in a way that encodes the label structure.

Once we obtained the embedding function f_ϕ, we use two auxiliary networks, learned end-to-end together with the meta-learner, to predict the shift and scale factors of the

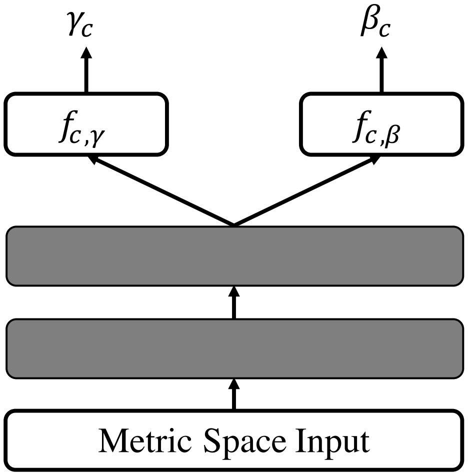


Figure 4.3: CBN shared architecture

convolutional feature map:

$$\hat{\gamma}_{i,c} = f_{c,\gamma}(f_\phi(\mathbf{x}_i)), \qquad \hat{\beta}_{i,c} = f_{c,\beta}(f_\phi(\mathbf{x}_i)). \tag{4.3}$$

Having computed $\hat{\gamma}_{i,c}$ and $\hat{\beta}_{i,c}$, Conditional Batch Normalization (CBN) is applied as follows:

$$\mathrm{CBN}(\mathbf{R}_{i,c}|\hat{\gamma}_{i,c},\hat{\beta}_{i,c}) = \hat{\gamma}_{i,c}\frac{\mathbf{R}_{i,c} - \mathbb{E}[\mathbf{R}_c]}{\sqrt{\mathrm{Var}[\mathbf{R}_c] + \epsilon}} + \hat{\beta}_{i,c}, \tag{4.4}$$

where $\mathbf{R}_{i,c}$ refers to the c^{th} feature map from the i^{th} example, ϵ is a small constant, β_c and γ_c are learnable parameters shared within a task. $\mathbb{E}[\mathbf{R}_c]$ and $\mathrm{Var}[\mathbf{R}_c]$ are batch mean and variance of $\mathbf{R}_c$.

It is worthwhile to note the effect of conditional transformation. The conditional bias transformation with $\hat{\beta}_{i,c}$ is analogous to concatenation-based conditioning where the conditional information is concatenated to the feature maps [Dumoulin et al., 2018]. The conditional scaling factor provides multiplicative interactions between the metric space and the feature maps to aggregate information. Furthermore, the conditional batch normalization operation is always followed by a rectified linear unit, which dynamically turns on or off the feature representation depending on whether the conditional output is greater than zero or not. The conditional transformation can be understood as a gating mechanism where the metric space's label information controls the feature representations.

4.2.3 Multitask learning of CBN

Although one can predict $\hat{\gamma}_c$ and $\hat{\beta}_c$ using two separate functions, we find it beneficial to use shared parameters as shown in Figure 4.3. The shared representations are more efficient at producing conditional transformations which also provide a strong inductive bias to help learning [Caruana, 1997].

4.2.4 Training Details

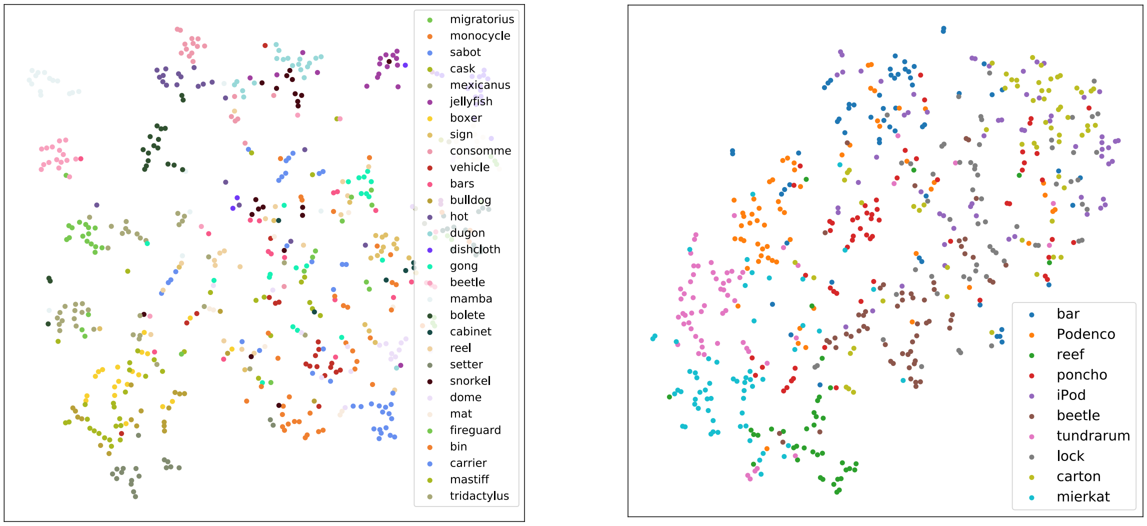
The base-learner (f_θ) is composed of 4 layers of 3×3 convolutions with a 4×4 skip connections from the input to the final convolutional layer. Each convolutional layer has 30 channels and is followed by CBN, ReLU and 2×2 max-pooling operations. The output of the final convolution is flattened and fed to a 1-layer dense classifier. For learning the embedding space (f_ϕ), we use the same residual network as Oreshkin et al. [2018]. For CBN functions (f_c), we use 3 dense layers with 30 hidden units each. Every layer is followed by a ReLU except for the last layer where no activation is used. For the meta-learner, we use MAML with 1 gradient step for 1-shot learning and 5 gradient steps for 5-shot learning. We use the Adam [Kingma and Ba, 2014] optimizer and clip the L2 norm of gradients with an upper bound of 5.

4.3 Experiments

We use *mini*ImageNet to evaluate the proposed Conditional class-Aware Meta-Learning algorithm. *mini*ImageNet [Vinyals et al., 2016] is composed of 84×84 colored images from 100 classes, with 600 examples in each class. We adopt the class split by Ravi and Larochelle [2016] that uses 64 classes for training, 16 for validation, and 20 for test. For N-way K-shot training, we randomly sample N classes from the meta-train classes each containing K examples for training and 20 examples for testing. At meta-testing time, we randomly sample 600 N-way K-shot tasks from the test classes.

4.3.1 Results

The results presented in Table 4.1 show that our proposed algorithm has comparable performance on the state-of-the-art *mini*ImageNet 5-way 1-shot classification task, and competitive results on the 5-way 5-shot task.



((a)): Randomly selected 30 training labels

((b)): Randomly selected 10 validation labels

Figure 4.4: t-SNE visualization of the learned metric space colored by category.

Table 4.1: *mini*ImageNet classification accuracy with 95% confidence intervals.

Model	5-way 1-shot	5-way 5-shot
Meta-Learner LSTM [Ravi and Larochelle, 2016]	43.44% ± 0.77%	60.60% ± 0.71%
Matching Networks [Vinyals et al., 2016]	46.6%	60.0%
Prototypical Network with Soft k-Means [Ren et al., 2018]	50.41% ± 0.31%	69.88% ± 0.20%
MetaNet [Munkhdalai and Yu, 2017]	49.21% ± 0.96%	–
TCML [Mishra et al., 2018]	55.71% ± 0.99%	68.88% ± 0.92%
adaResNet [Munkhdalai et al., 2018]	56.88% ± 0.62%	71.94 ± 0.57%
Cosine Classifier [Gidaris and Komodakis, 2018]	56.20% ± 0.86%	73.00% ± 0.64%
TADAM [Oreshkin et al., 2018]	58.5%	**76.7%**
LEO [Rusu et al., 2018]	**60.06% ± 0.05%**	75.72% ± 0.18%
MAML [Finn et al., 2017a]	48.7% ± 1.84%	63.11% ± 0.92%
MAML on our architecture	48.26% ± 1.04%	64.25% ± 0.78%
Prototypical Network [Snell et al., 2017b]	49.42% ± 0.78%	68.2% ± 0.66%
Prototypical Network on our metric space	55.96% ± 0.91%	71.64% ± 0.70%
CAML (with multitask learning alone)	52.56% ± 0.83%	71.35% ± 1.13%
CAML (with class-aware grouping alone)	55.28% ± 0.90%	71.14% ± 0.81%
CAML (full model)	**59.23% ± 0.99%**	72.35% ± 0.71%

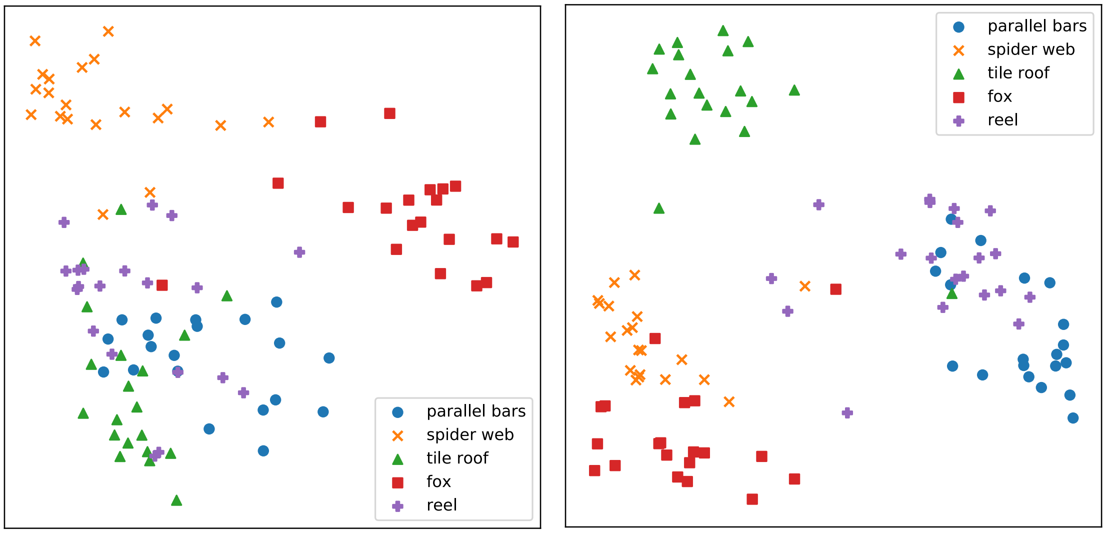
Figure 4.4 shows the t-SNE plot of the learned metric space for both meta-train and meta-validation classes. As seen in Figure 4.4, examples from the meta-validation set tend to form clusters consistent with their class membership, even though the metric space is not trained on these classes. For example, “mierkat", “tundrarum" and “podenco" are all animals and they are clustered close together.

The first main baseline we report is MAML. CAML improves upon MAML by about 10% on both 1-shot and 5-shot tasks. This means incorporating class dependencies in the form of a metric space can greatly facilitate gradient-based meta-learning. We also compare with MAML using our base-learner architecture equipped with skip connections from the input to the last convolutional layer. MAML trained with our base-learner's architecture yields similar performance as the original MAML, suggesting the improvement is resulted from the proposed CAML framework, rather than changes in the base-learner's architecture.

The second baseline we use is the prototypical network. We measure the classification ability of our metric space using prototypical network as a classifier, shown in Table 4.1 (Prototypical Network in our metric space). These results suggest that making predictions on the metric space alone is inferior to CAML.This can be explained by CAML's ability to fast-adapt representations even when the metric space does not provide good separations. We also find that CAML has larger improvements in 1-shot tasks than 5-shot ones. This is because, in 1-shot learning, metric-based methods estimate class representations from a single example, making it difficult to provide a robust class estimation.

Better ways to learn the dynamic bias could lead to improved generalization. TADAM [Oreshkin et al., 2018] and LEO [Rusu et al., 2018] outperform the proposed CAML approach. Like CAML, TADAM also uses conditional information, but TADAM modulates the representation of a metric space instead of the feature space. LEO learns a data-dependent mapping to latent representation.

Recent research on task normalization [Bronskill et al., 2020] highlights the limitations of transductive batch normalization. Extending conditional batch normalization to conditional task normalization is a promising direction of future work.



((a)): Before conditional transformation **((b)):** After conditional transformation

Figure 4.5: PCA visualization of feature maps from the last convolutional layer colored by category.

4.3.2 The effect of conditional transformation

We compare activations before and after the conditional transformation to better understand how conditional transformation modulates the feature representations. Figure 4.5 shows the PCA projections of the last convolutional layer in the base-learner. We observe in Figure 4.4(a) that, before conditional transformation, examples from three classes ("parallel bars", "tile roof" and "reel") are mixed together. In Figure 4.4(b), after the conditional transformation is applied, one of the previously cluttered classes ("tile roof") become separated from the rest classes. This confirms that metric space can alleviate the difficulty in few-shot learning by means of conditional transformations.

We undertake ablation studies to show the impact of multitask learning and class-aware grouping. Empirical results in Table 4.1 suggest that, while 1-shot learning is sensitive to multitask learning and class-aware grouping, 5-shot learning is not affected by those techniques. This is owing to a lack of statistical strength in 1-shot learning, which requires more explicit guidance in the training procedure. This means exploiting metric-based channel mean and variance can provide valuable information to improve meta-learning.

4.4 Limitations

One limitation of the proposed approach is in its inability to quantify uncertainties in the metric space. The metric space is deterministic and is not able to distinguish different levels of hierarchies in the label space. Note that the metric space is not performing hierarchical classification [Silla and Freitas, 2011] on a tree or direct acyclic graph in the label space; instead, the metric space intends to indicate differences in the representational structures between classes to inform the learning of various tasks. It would be more informative if the conditional transformation is also guided by the degree of uncertainty in the metric space.

Another limitation is in the lack of evaluation on "meta-overfitting." Triantafillou et al. [2019] discovered that the episodic boostrapping-style training strategy could overfit on one dataset and does not generalize well to another dataset. This arises from generalizing to tasks within the same dataset to generalizing to tasks between different datasets. More thorough evaluation is needed to further examine the generalization of the proposed approach.

4.5 Conclusions

In this chapter, we propose Conditional class-Aware Meta-Learning (CAML) that incorporates class information by means of an embedding space to conditionally modulate representations of the base-learner. By conditionally transforming the intermediate representations of the base-learner, our goal is to reshape the representation with a global sense of class structure. Experiments reveal that the proposed conditional transformation can modulate the convolutional feature maps towards a more disentangled representation. We also introduce class-aware grouping to address a lack of statistical strength in few-shot learning. The proposed approach obtains competitive performance on 5-way 1-shot *mini*ImageNet benchmark.

Chapter 5

Task Adaptive Metric Space for Medium-Shot Medical Image Classification

5.1 Introduction

Learning new concepts from a small number of examples is an essential ability of human cognition. While deep learning models typically require a large number of labeled examples, datasets in the medical imaging domain tend to have a limited number of training examples. Efforts towards more sample-efficient training procedures address this data hindrance, thereby enabling the wider use of deep learning techniques for medical applications.

Meta-learning [Bengio et al., 1992; Mitchell and Thrun, 1993; Schmidhuber, 1987; Vilalta and Drissi, 2002], or "learning to learn", is an important approach to few-shot classification. The meta-learner is trained across many tasks to acquire meta-knowledge with the goal of learning a new task with few labeled examples. Despite the recent popularity of meta-learning, we find two main limitations in applying meta-learning to medical image classification tasks.

The first limitation is the discrepancy of the number of training examples between few-shot tasks and medical datasets. While in the literature meta-learning mostly deals with tasks with only a few training examples, i.e., five or fewer, datasets in the medical domain tend to have tens to a few hundred labeled examples. This discrepancy necessitates the extension from few-shot learning to *medium-shot* for more realistic evaluations on the medical domain.

The second limitation is the lack of meta-learning evaluation procedures on medical datasets. We propose to evaluate representative meta-learning methods under different amounts of data per class, to better understand their generalization properties. We choose key advances in meta-learning—*gradient-based* [Finn et al., 2017a] and *metric-based* [Snell et al., 2017b] methods—to establish the baseline performances.

We empirically evaluate and analyze the baseline meta-learning methods through the lens of bias-variance tradeoff. Our analysis suggests gradient-based methods tend to overfit few-shot datasets while metric-based methods tend to underfit medium-shot datasets. To get the best of both worlds for bias-variance equilibrium, we propose Task Adaptive Metric Space (TAMS) that uses gradient-based fine-tuning to adjust parameters of the metric space so that distances between examples in the medical dataset can better reflect their semantics. We show that our proposed model outperforms gradient-based and metric-based meta-learning models on a medical image classification task. We report visualizations on the metric space that validates the impacts of metric adaptation.

Our main contributions are three-fold: (1) We propose *medium-shot learning* that aligns meta-learning with realistic situations of medical image classification. (2) We establish baseline evaluation procedures to evaluate meta-learners on various situations to better understand their generalization properties. (3) Through bias-variance analysis, we propose a new meta-learning method—Task Adaptive Metric Space—that takes advantage of both gradient-based and metric-based methods. We show that TAMS outperforms the meta-learning baselines.

5.2 Background and Motivation

Medium-shot Learning for Medical Data. Medical datasets raise many practical challenges especially because they tend to have tens to a few hundred labeled examples [Litjens et al., 2017]. The size of medical datasets does not align with the prevalent approaches in few-shot learning that only focus on five or fewer training examples per class. For this reason, we propose *medium-shot* learning with the intention of more realistic assessments in the medical domain. We define "medium-shot" as classification tasks with tens to a few hundred labeled examples each class.

Overview of Meta-learning. The goal of meta-learning is to acquire meta-knowledge from many tasks to help better learn a new task. In recent years, meta-learning has become an important approach for few-shot learning [Vinyals et al., 2016]. A meta-learning system first aims to *meta-train* the meta-learner from $\mathscr{D}_{\text{meta}-\text{train}}$, which is composed of many classification tasks $\mathcal{D}_i \in \mathscr{D}_{\text{meta}-\text{train}}$ and each task can be further split into a training and test set $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) \in \mathcal{D}$. Once the meta-learner is trained, we *meta-test* a new task $\mathscr{D}_{\text{meta-test}}$ using the meta-learned inductive bias in the form of weight initializations or a metric space. We discuss two representative meta-learning methods through the lens of bias-variance tradeoff [Friedman et al., 2001].

Gradient-based Meta-learning. Gradient-based methods, such as Model-Agnostic Meta-Learning (MAML) [Finn et al., 2017a], aim to optimize representations for fast adaptation across many tasks. The meta-learned knowledge is encoded through parameter initializations to support fast adaptation when fine-tuned on $\mathscr{D}_{\text{meta-test}}$. In the medical domain, MAML has been explored in breast screening classification to meta-learn initializations with curriculum learning [Maicas et al., 2018]. Gradient-based methods tend to have a low bias and high variance. They can take advantage of more data to better adapt model representations towards a new task. However, gradient-based methods are more complex and may overfit because the fine-tuning procedure updates all parameters to adapt to $\mathscr{D}_{\text{meta-test}}$. Recent work on proximal regularization [Rajeswaran et al., 2019] aims to reduce over-fitting by increasing the strength of the prior over the data via the regularization on meta-parameters. However, due to the large meta-hypothesis space, stronger regularization is needed to further reduce the variance of MAML-based approaches.

Metric-based Meta-learning. Metric-based methods, such as Prototypical Networks [Snell et al., 2017b], learn a metric space across many tasks such that distances between examples are semantically meaningful. The metric space is parameterized with a neural network meta-trained on $\mathscr{D}_{\text{meta-train}}$. At meta-test time, each class is represented by a prototype in the metric space, and classification labels are assigned with distance-based inference. Metric-based methods tend to have high bias and low variance. An advantage of metric-based methods is their non-parametric classification procedure that prevents overfitting on a few examples, but their static metric space may result in underfiting when more data is available.

The Best of Both Worlds. The bias-variance tradeoff [Friedman et al., 2001] tells us that the gradient-based and metric-based methods complement each other with different amounts of training data. In TAMS, we fuse these methods and demonstrate the improved generalization under several setups, which is congruent with out intuition.

5.3 Task Adaptive Metric Space for Improved Medium-shot Generalization

We propose Task Adaptive Metric Space (TAMS) to exploit the inherent expressiveness of gradient-based methods and the non-parametric property of metric-based methods. We note that TAMS is built upon a metric space that is meta-trained using Prototypical Networks [Snell et al., 2017b] on $\mathscr{D}_{\text{meta}-\text{train}}$, and we focus on how to adapt the meta-learned metric space to better fit the medical classification task $(\mathcal{D}_{\text{train}},\mathcal{D}_{\text{test}}) \in \mathscr{D}_{\text{meta}-\text{test}}$. The metric space f_ϕ is initialized with meta-learned parameters from $\mathscr{D}_{\text{meta}-\text{train}}$ prior to task adaptive fine-tuning on $\mathscr{D}_{\text{meta}-\text{test}}$.

Partition the Training Set $\mathcal{D}_{\text{train}}$. Prototypical networks use all training data to construct prototypes and the training examples are never evaluated against their labels, hence *cannot* provide a loss function to estimate the quality of the metric space. However, for metric fine-tuning, we need to quantify the extent to which the metric space fits $\mathcal{D}_{\text{train}}$ to penalize errors and to improve the metric space. To address this, we propose to randomly partition the training examples into disjoint sets: $\mathcal{D}_{\text{train}}^{\text{prototype}}$ for computing a prototype for each class and $\mathcal{D}_{\text{train}}^{\text{predict}}$ for assessing the quality of the prototypes. The metric space is adapted to a new task by maximizing the likelihood that the data are generated by the prototypes. This provides an evaluation measure that estimates how suitable the metric is for the training data, which further allows us to fine-tune the metric space to better represent the medical dataset.

Construct Prototypes from $\mathcal{D}_{\text{train}}^{\text{prototype}}$. Prototypical networks reduce the training data into points $\mathbf{c}_t$ on the metric space as prototypical representations of class t. We only use $\mathcal{D}_{\text{train}}^{\text{prototype}}$ to construct $\mathbf{c}_t = \frac{1}{K}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_{\text{train}}^{\text{prototype}}} \mathbb{1}_{\{y_i=t\}} f_\phi(\mathbf{x}_i)$, where K denotes the number of examples for class t, $\mathbb{1}_{\{y_i=t\}}$ denotes an indicator function of y_i which takes value 1 when $y_i = t$ and 0 otherwise. The metric space is parameterized by ϕ in the form of a convolutional or a residual network.

Make Predictions on $\mathcal{D}_{\text{train}}^{\text{predict}}$. We evaluate the prototypes $\mathbf{c}_t$ on $\mathcal{D}_{\text{train}}^{\text{predict}}$ to estimate the quality of the metric space for the medical image classification task. To obtain the prediction labels, each example $\mathbf{x} \in \mathcal{D}_{\text{train}}^{\text{predict}}$ is first mapped to $f_\phi(\mathbf{x})$ on the metric space, then being classified based on its relative distances with each prototype $\mathbf{c}_t$, according to the distance function d:

$$p(y=t|\mathbf{x})=\frac{\exp(-d(f_\phi(\mathbf{x}),\mathbf{c}_t)))}{\sum_{t'}\exp(-d(f_\phi(\mathbf{x}),\mathbf{c}_{t'}))}. \tag{5.1}$$

Loss for Metric Fine-tuning. We use the cross-entropy loss to evaluate the predictions on the medical dataset $\mathcal{D}_{\text{train}}^{\text{predict}}$. The loss can be reinterpreted as a maximum-likelihood objective that aims at minimizing the distance between an example and its corresponding class centroid while maximizing its distance to other class centroids [Snell et al., 2017b].

Make Predictions on $\mathcal{D}_{\text{test}}$. With the task-adapted metric space on $\mathcal{D}_{\text{train}}$, we can make predictions on the testing data $\mathcal{D}_{\text{test}}$. Unlike the metric fine-tuning step that constructs prototypes from $\mathcal{D}_{\text{train}}^{\text{prototype}}$, in this prediction step, we use all training examples in $\mathcal{D}_{\text{train}}$ to represent the prototypes. Note that two types of prototypes are constructed: (i) the first set of prototypes are derived from $\mathcal{D}_{\text{train}}^{\text{prototype}}$ to measure how well the metric space represents data from the new task. (ii) the final prototypes for inferring new examples are calculated on the full training set $\mathcal{D}_{\text{train}}$ after the metric space has been adapted to the new task. We use the same distance-based classifier as Eq. (5.1) on $\mathcal{D}_{\text{test}}$ as final predictions on the test data. The prototype-based classifier differs from the k-nearest neighbor approach in that the prototypes are learned in a supervised fashion by adjusting the parameters of the metric space based on all training examples. In contrast, in the k-nearest neighbor approach, classes are determined by a memory-based procedure where only a subset of the training data contributes to each example's predictions.

To summarize, TAMS adapts a meta-learned metric space to better represent medical data. This improves medium-shot generalization in spite of domain difference between $\mathscr{D}_{\text{meta-train}}$ and $\mathscr{D}_{\text{meta-test}}$ which would otherwise be challenging for current meta-learning methods. TAMS constitutes a natural bridge between gradient-based and metric-based methods. In few-shot situations, the non-parametric classification procedure prevents our model from overfitting. In medium-shot situations, fine-tuning the metric space exploits the expressive richness of gradient-based methods to fit the target medical classification task better.

5.4 Empirical Results

5.4.1 Data and Evaluation Setup

The experiments first pre-train a model with meta-learning or transfer learning on $\mathscr{D}_{\mathrm{meta-train}}$, then evaluate the generalization properties on the target meta-test dataset $\mathscr{D}_{\mathrm{meta-test}}$, i.e., the target medical classification task.

Meta-train Dataset. We use *mini*ImageNet [Ravi and Larochelle, 2016; Vinyals et al., 2016]—a standard meta-learning dataset—as $\mathscr{D}_{\mathrm{meta-train}}$. Because the goal of this chapter is to propose TAMS and evaluate it against other meta-learning baselines in unseen domains, we do not incorporate other medical datasets in meta-training.

Meta-test Dataset. We use Optical Coherence Tomography (OCT) [Kermany et al., 2018] as $\mathscr{D}_{\mathrm{meta-test}}$. OCT aims to classify each image into one of the four classes: NORMAL, CNV, DME, DRUSEN. We created our train-test split that consists of up to 250 examples per class as $\mathcal{D}_{\mathrm{train}}$, and 250 examples per class as $\mathcal{D}_{\mathrm{test}}$. We include more details about data statistics and preprocessing in the supplementary material.

Evaluation Setup. To examine the impact of training data on model performance, we vary the number of training examples and use 1 to 250 examples per class as training data $\mathcal{D}_{\mathrm{train}}$. All experiments are repeated ten times with controlled random seeds. In terms of model architectures, we first evaluate different methods with a standard meta-learning architecture: "conv4"—a 4-layer convolutional network. We then experiment with "resnet12", a 12-layer residual network, to assess the impact of added model capacity.

Baselines. We use the following transfer learning and meta-learning baselines:

1. "Scratch": a basic model that trains $\mathcal{D}_{\mathrm{train}}$ from random initialization.

2. "Transfer": a transfer learning baseline where the model is pre-trained on *mini*ImageNet, and then fine-tuned on $\mathcal{D}_{\mathrm{train}}$.

3. "MAML": a gradient-based meta-learning baseline that meta-trains MAML parameters on *mini*ImageNet, and fine-tunes on $\mathcal{D}_{\mathrm{train}}$.

4. "Proto": a metric-based prototypical network meta-trained on *mini*ImageNet.

Table 5.1: OCT test accuracy with "conv4" architecture (%)

shots		16 gradient steps				32 gradient steps			
	Proto	Scratch	Transfer	MAML	TAMS	Scratch	Transfer	MAML	TAMS
1	**28.18**	25.72	26.97	27.80	—	25.88	26.93	27.84	—
2	33.14	27.58	28.93	33.63	32.13	27.78	29.05	**34.04**	32.02
4	35.51	26.84	30.31	33.12	37.34	27.62	31.20	33.93	**37.80**
8	40.93	29.37	34.52	38.00	**42.73**	30.11	36.27	38.50	42.49
16	41.53	32.83	40.27	42.29	47.55	34.75	41.98	44.27	**48.07**
32	46.38	35.72	41.97	43.23	53.71	38.25	45.32	45.45	**53.74**
64	48.20	38.84	44.49	44.91	56.30	42.45	48.48	48.59	**57.51**
128	49.17	43.45	47.95	48.68	57.53	48.62	52.84	54.35	**60.55**
250	49.80	46.89	50.28	50.00	60.83	53.72	57.34	57.65	**63.57**

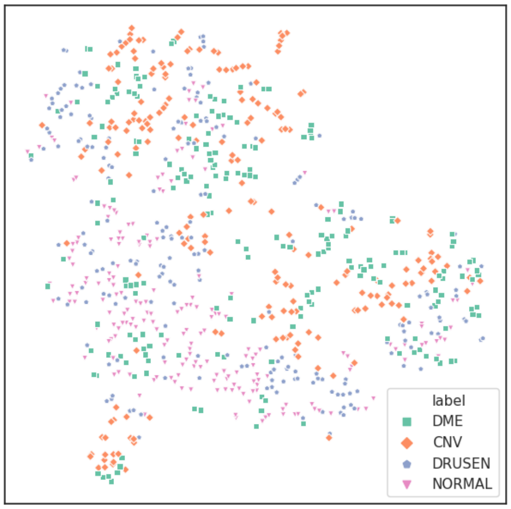


Figure 5.1: t-SNE visualization of sampled test data before adaptation

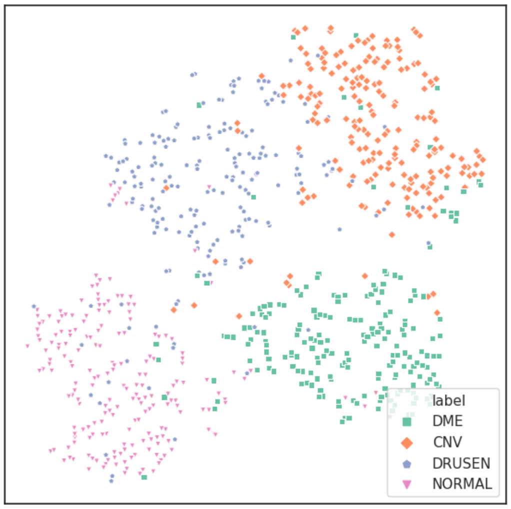


Figure 5.2: t-SNE visualization of sampled test data after adaptation

5.4.2 Results and Discussions

Table 5.1 summarizes the test accuracies on OCT with various shots per class. All models use "conv4" architecture, and the accuracy is averaged over ten runs.

Transfer Learning and Gradient-based Meta-learning. We first investigate the impact of transfer learning and gradient-based meta-learning by comparing "Scratch" with "Transfer" and "MAML". The three methods only differ in parameter initialization: "Scratch" is randomly initialized, "Transfer" is initialized from a pre-trained classifier, and "MAML" is initialized from meta-learned parameters. In Table 5.1, under the same number of gradient steps, we find "Transfer" and "MAML" perform better than "Scratch" because of better parameter initializations. We also find “MAML” works better than “Scratch” and “Transfer” when shots are less than 32 because “MAML” is optimized for few-shot learning.

The Bias-variance Tradeoff. As a critical motivation to TAMS, we highlight the bias-variance tradeoff by comparing “MAML” with “Transfer” and “Proto”. In few-shot scenarios, as shown in Table 5.1, “Proto” outperforms ‘Transfer” and “MAML” under 1, 4 and 8 shots, even after “Transfer” and “MAML” are fine-tuned with more gradient steps. This suggests gradient-based methods tend to overfit in few-shot. However, as we increase the number of training examples, we find that “MAML” with 32 gradient steps outperforms “Proto” at 16, 64, 128 and 250 shots, suggesting metric-based methods tend to underfit in medium-shot.

The Effect of Metric Adaptation. To validate the effectiveness of our proposed method, we compare TAMS with all baseline methods under the same number of gradient steps. Table 5.1 shows that TAMS achieves the best test accuracy in most cases. Take 128-shot classification as an example, under 32 gradient steps, TAMS outperforms MAML by 6% and outperforms Proto by 10%. This suggests TAMS achieves better bias-variance equilibrium, alleviating the overfitting of gradient-based methods while preventing underfitting of metric-based methods. Figure 5.1 and 5.2 shows testing examples projected on the metric space before and after metric adaptation. While examples from different classes are mixed before metric adaptation, TAMS results in well-separated clusters that reflect their labels. This confirms that TAMS is capable of adjusting parameters of the metric space to better represent examples in semantically meaningful ways. Furthermore, TAMS is efficient to train as it requires a small number of gradient steps.

The Impact of Metric Adaptation Steps. Figure 5.3 shows the impact of metric adaptation steps on the test accuracy. We find that a few adaptations steps are sufficient in few-shot, but more adaptations steps are needed in medium-shot.

The Impact of Model Capacity. As an ablation study, we investigate the impact of model capacity on different meta-learners. We have the following findings from Figure 5.4: (i) With respect to metric adaptation, TAMS improves dramatically as the

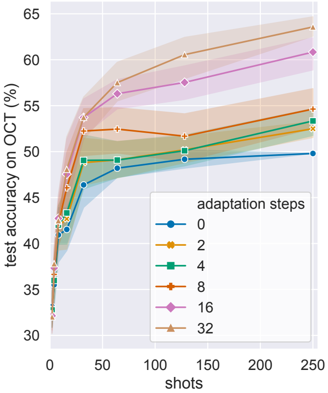


Figure 5.3: Metric adaptation steps.

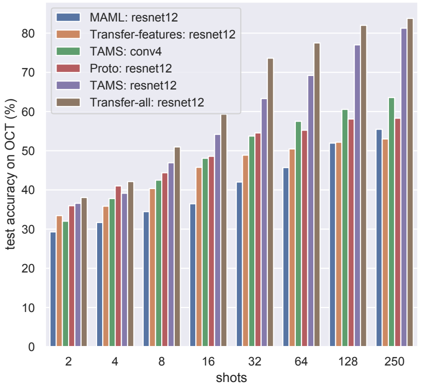


Figure 5.4: Impact of model architecture.

model capacity increases from “conv4“ to “resnet12“. We also highlight the improved performance over “Proto” and “MAML” brought by our proposed TAMS. (ii) Concerning different transfer learning approaches, we find transfer learning without updating the features on the new task—“Transfer-features”—does not work well. This difference can be attributed to the domain difference between $\mathscr{D}_{\mathrm{meta-train}}$ and $\mathscr{D}_{\mathrm{meta-test}}$ and the need for the model to adjust its feature representations to better fit $\mathscr{D}_{\mathrm{meta-test}}$. (iii) We find the transfer learning method “Transfer-all”, that fine-tunes both feature representations and the classifier, performs best with “resnet12”. Despite this, we highlight that the proposed TAMS greatly outperforms other meta-learning methods indicating the exciting potential of metric-adaptive meta-learning. We believe better metric loss functions, such as contrastive loss [Koch et al., 2015] and triplet loss [Schroff et al., 2015], could further improve the performance of TAMS for better medium-shot medical image classification. Due to page limits, we include more empirical results in supplementary materials.

5.5 Limitations

The main limitation of the empirical study is that the proposed model is not evaluated on large-scale meta-learned representations, such as ImageNet, due to computational limitations. This makes it difficult to fully evaluate the effectiveness between weight transfer and learning-to-learn in real-world applications.

The main limitation of the proposed approach is in the definitions of the task space. It is sometimes difficult to gather a large number of class labels to sample tasks from. Self-supervised learning-to-learn approaches are promising directions to resolve this problem.

5.6 Conclusions

With this chapter, we hope to draw the attention of the medical imaging community to the rich field of meta-learning, which offers feasible solutions to situations of the limited training examples that the field is often faced with. To better evaluate realistic situations in the medical domain, we extend few-shot learning to medium-shot and establish a baseline procedure that aims to evaluate representative meta-learning algorithms on various amounts of training data. This serves as a baseline for future explorations using meta-learning in the medical domain. Through bias-variance analysis, we identify complementary roles of gradient-based and metric-based meta-learning and propose to fuse the best of both methods into Task Adaptive Metric Space. Our experiments reveal that the proposed metric adaptation method can adjust the metric space to better reflect examples of a new medical classification task.

Chapter 6

Conclusion and Future Research

6.1 Conclusion

In this thesis, we first study unsupervised domain adaptation, an emerging field of semi-supervised learning that aims to address domain shift based on labeled data in the source domain together with unlabeled data in the target domain. We propose implicit class-conditioned domain alignment to address between-domain class distribution shift. We provide a theoretical analysis to justify the proposed method by decomposing the empirical domain divergence into class-aligned and class-misaligned divergence, and we show that class-misaligned divergence is detrimental to domain adaptation. We identify a domain discriminator shortcut function that interferes with adversarial domain adaptation because the model could bypass the optimization for domain-invariant representations, but rather optimize for a shortcut function that is independent of the covariate contributing to the domain difference. We show that our method offers consistent improvements for different adversarial adaptation algorithms: both DANN and MDD. We also design extensive experiments to demonstrate the effectiveness of the proposed method by simulating various degrees of between-domain class distribution shift. The empirical results reveal that the proposed method improves other domain adaptation algorithms regardless of the degree or the type of distribution shift.

We also propose two meta-learning methods to bridge the gap between gradient-and-metric-based methods. The first proposal is Conditional class-Aware Meta-Learning where we introduce a metric space trained to encode regularities of the label structure so as to impose global class dependencies to the model. The second proposal is to extend few-shot learning to few-to-medium-shot learning. This is motivated by the discrepancy of the number of training examples between few-shot tasks and real-world medical datasets. While in the literature meta-learning mostly deals with tasks with only a few training examples, i.e., five or fewer, datasets in the medical domain tend to have tens to a few hundred labeled examples. We empirically evaluate and analyze the baseline meta-learning methods through the lens of bias-variance tradeoff on medical datasets. Our analysis suggests gradient-based methods tend to overfit few-shot datasets while metric-based methods tend to underfit medium-shot datasets. We propose Task Adaptive Metric Space that uses gradient-based fine-tuning to adjust parameters of the metric space to provide more flexibility to metric-based methods. Our experiments reveal that the proposed metric adaptation method can adjust the metric space to better reflect examples of a new medical classification task. Regarding the difference between the two proposed approaches, CAML learns to impose a dynamic bias on a family of tasks. In contrast, TAMS learns to adapt a metric space to a different task. For this reason, if a new task is more closely related to the family of meta-training tasks, CAML should be preferred to TAMS. On the other hand, if the new task has low relation with the meta-training tasks, the meta-learned dynamic bias might not be generalizable; therefore, TAMS should be preferred over CAML.

6.2 Future Research

6.2.1 Meta-learning Domain Adaptation

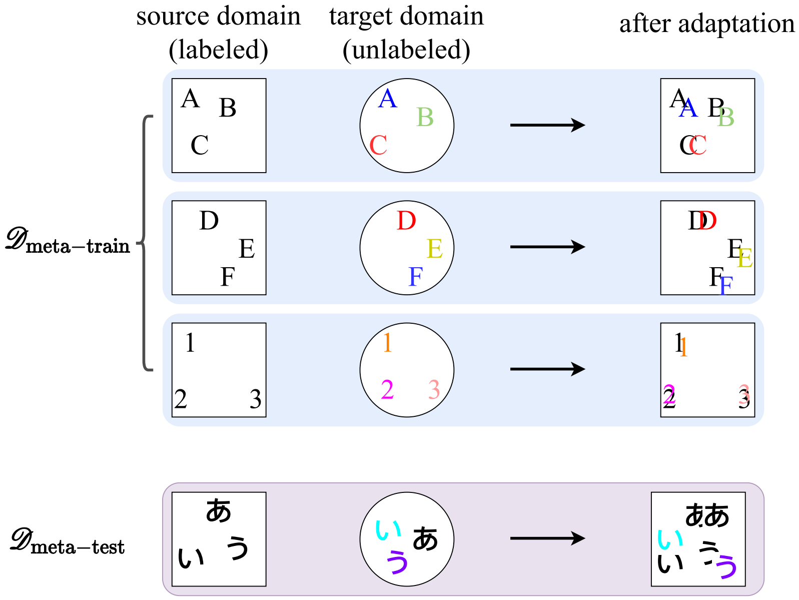


Figure 6.1: meta-learning domain adaptation: each row represents an unsupervised domain adaptation task where both the classification task (distinguishing different characters) and domains (colors) are different. The goal is to generalize from $\mathcal{D}_{\text{meta-train}}$ to $\mathcal{D}_{\text{meta-test}}$.

Although much of the work in domain adaptation is on the same task space, it is appealing to be able to also generalize to new domains together with new tasks. This is a more general form of meta-learning where $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are the same task but in different domains, i.e., each inner task itself is a domain adaptation task, where $\mathcal{D}_{\text{train}}$ is labeled and $\mathcal{D}_{\text{test}}$ is unlabeled. The goal for this setup is to generalize to a new task as well as an unseen domain, i.e., domain-adaptive cross-task generalization. This is depicted in Figure 6.1 where $\mathscr{D}_{\text{meta}-\text{train}}$ comprises a set of different domain adaptation tasks on both different label space and different domain pairs. The assumption is that the underlying factors that contribute to the domain difference, such as different colors, can be learned and decomposed into different factors through $\mathscr{D}_{\text{meta}-\text{train}}$ and generalize to $\mathscr{D}_{\text{meta}-\text{train}}$.

As a concrete example, the same MRI scanner can be configured differently for different hospitals or patients, where each configuration can be treated as a different domain. Although it is feasible to adapt from one configuration to another directly, the knowledge between various adaptation procedures, while adapting between many configurations, are not shared to facilitate better adaptation to new configurations. meta-learning domain adaptation enables the model to reuse previous domain adaptation experiences to better adapt to new domains. In addition, meta-learning domain adaptation makes domain adaptation more efficient by re-using the domain knowledge across different tasks: adapting a liver segmentation model from domain A to domain B should make adapting a lung segmentation model easier.

One potential approach is to use Reptile [Nichol et al., 2018]—a first-order MAML style approach—that randomly samples domain adaptation pairs and uses the loss from target domain to update the meta parameters until the model converges. For meta-testing on a new task with a different domain pair, we can use the meta-learned parameters as initialization. However, preliminary exploratory efforts suggest first-order approaches like Reptile does not converge well with adversarial domain adaptation approaches. The main challenge is that the task-specific gradient information is highly sensitive to meta-aggregation in the form of simple averaging, and more efforts need to be put in developing adversarial friendly approaches to meta-learning.

6.2.2 Privacy-Preserving Domain Adaptation

Privacy has become a major concern for both individuals and institutions when it comes to data sharing to develop effective applied machine learning models. This makes it difficult to make use of collaborative learning where data are stored in isolated "islands" and it is prohibitive to transfer data to a central server for joint training due to privacy concerns. This is especially relevant for domain adaptation because the data of different domains typically reside in different physical environments.

Three types of approaches can be used to address the privacy concern for collaborative domain adaptation: differential privacy [Dwork et al., 2006], federated learning [Konečnỳ et al., 2015], and hypothesis transfer [Kuzborskij and Orabona, 2013]. The differential privacy approach protects sensitive informaton by sharing randomized aggregated information, as opposed to individual data samples. In deep learning, training methods that respect differential privacy have been proposed with the ability to control the influence of the training data through stochastic optimiazation [Abadi et al., 2016]. The second approach is federated learning where models are trained in parallel without exchanging data between different clients. Instead, the models only exchange parameters and use some aggregation algorithms to aggregation the knowledge of distributed clients into a centralized server. The third approach is hypothesis transfer where transfer learning happens at the hypothesis level in the form of parameters, rather than instance-level transfer learning. The goal is to adapt the hypothesis trained on the source domain to a set of unlabeled target domain data without access to the source domain data.

The proposed implicit class-conditional domain alignment in Chapter 3 can be integrated into federated learning and hypothesis transfer by employing the class-aligned sampling strategy protect against membership inference attacks from the marginal class distribution. Furthermore, implicit alignment can be reformulated as class-conditioned parametric bootstrap where adversarial examples are sampled to learn domain invariance while minimizing the risk of adversarial attacks by learning smooth decision boundaries.

6.3 Grounded meta-learning

The dominant paradigm in current natural language understanding and computer vision is to learning statistical patterns from data, such as language models and self-supervised image representation learning. These approaches typically require large amounts of training data and suffer from spurious correlations in the dataset due to the lack of high-level semantic understanding.

Visually-grounded language learning has emerged into a new research direction that aims to learn language representations in multimodal and interactive environments where the language is grounded by its interactions with the environments. This is similar to how children acquires vocabulary through interactions with concepts in the real world [Gopnik and Meltzoff, 1984]. It is therefore promising to bridge computer vision, natural language understanding, and interactions with the real world environments under the framework of meta-learning. Pioneering work in this direction [Co-Reyes et al., 2018] shows that language-guided policy learning in the form of iterative language corrections can improve the efficiency of simulated navigation and manipulation tasks. Another example in this direction is to use simulation engines to help learn the intuitive physics of the world [Battaglia et al., 2013]. Intuitive physics links perception with higher cognition for abstract concepts understanding and can be used as an informative prior for visual understanding. Grounded meta-learning can help resolve the long-term problems faced by current natural language understanding and computer vision tasks [Hermann et al., 2017].

The proposed class-aware meta-learning in Chapter 4 can be generalized to context-aware [Silver et al., 2008] meta-learning where the grounded language information can be used as an conditional input to facilitate visual reasoning. In a similar way, visual information can be used as conditional input while training word embeddings through language models. The conditional embedding are context dependent and less ambiguous.

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

John Aldrich et al. Ra fisher and the making of maximum likelihood 1912-1922. *Statistical science*, 12(3):162–176, 1997.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

Robert B Ash, B Robert, Catherine A Doleans-Dade, and A Catherine. *Probability and measure theory*. Academic Press, 2000.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas, 1992.

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

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Paula Branco, Luís Torgo, and Rita P Ribeiro. Metautil: Meta learning for utility maximization in regression. In *International Conference on Discovery Science*, pages 129–143. Springer, 2018.

Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at . . . , 1997.

John Bronskill, Jonathan Gordon, James Requeima, Sebastian Nowozin, and Richard Turner. Tasknorm: Rethinking batch normalization for meta-learning. In *International Conference on Machine Learning*, pages 1153–1164. PMLR, 2020.

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

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

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

Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2090–2099, 2019b.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *Advances in neural information processing systems*, pages 2456–2464, 2011.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1081–1090, 2019c.

Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

John D Co-Reyes, Abhishek Gupta, Suvansh Sanjeev, Nick Altieri, Jacob Andreas, John DeNero, Pieter Abbeel, and Sergey Levine. Guiding policies with language via meta-learning. *arXiv preprint arXiv:1811.07882*, 2018.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Nanqing Dong and Eric P Xing. Domain adaption in one-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 573–588. Springer, 2018.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A LEARNED REPRESENTATION FOR ARTISTIC STYLE. *ICLR*, 2017. URL https://arxiv.org/pdf/1610.07629.pdf.

Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 3(7):e11, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *arXiv preprint arXiv:1710.11622*, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017a.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368, 2017b.

S Fralick. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, 13(1):57–64, 1967.

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

A Gammerman, V Vapnik, and VI Vovk. Learning by transduction. 1998.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1): 2096–2030, 2016.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

Faustino Gomez and Jürgen Schmidhuber. Evolving modular fast-weight networks for control. In *International Conference on Artificial Neural Networks*, pages 383–389. Springer, 2005.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Alison Gopnik and Andrew N Meltzoff. Semantic and cognitive development in 15-to 21-month-old children. *Journal of child language*, 11(3):495–513, 1984.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

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

James J Heckman. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161, 1979.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2007.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 264–271, 2007.

Xiang Jiang, Erico N de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L Silver, and Stan Matwin. Trajectorynet: An embedded gps trajectory representation for point-based classification using recurrent neural networks. *arXiv preprint arXiv:1705.02636*, 2017.

Xiang Jiang, Mohammad Havaei, Farshid Varno, Gabriel Chartrand, Nicolas Chapados, and Stan Matwin. Learning to learn with conditional class dependencies. In *International Conference on Learning Representations*, 2018.

Xiang Jiang, Liqiang Ding, Mohammad Havaei, Andrew Jesson, and Stan Matwin. Task adaptive metric space for medium-shot medical image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 147–155. Springer, 2019.

Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. *International Conference on Machine learning*, 2020.

Marc Kac and AJF Siegert. An explicit representation of a stationary gaussian process. *The Annals of Mathematical Statistics*, 18(3):438–442, 1947.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

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

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

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

Wouter Marco Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, pages 942–950, 2013.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Ke Li and Jitendra Malik. Learning to optimize neural nets. *arXiv preprint arXiv:1703.00441*, 2017.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2975–2984, 2019a.

Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Exploring uncertainty in pseudo-label guided unsupervised domain adaptation. *Pattern Recognition*, 96:106996, 2019b.

Omri Lifshitz and Lior Wolf. A sample selection approach for universal domain adaptation. *arXiv preprint arXiv:2001.05071*, 2020.

Charles X. Ling and Victor S. Sheng. *Cost-Sensitive Learning*, pages 231–235. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_181. URL https://doi.org/10.1007/978-0-387-30164-8_181.

Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*, 2018.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 97–105. JMLR. org, 2015.

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

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

Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in Neural Information Processing Systems*, pages 165–177, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

Gabriel Maicas, Andrew P Bradley, Jacinto C Nascimento, Ian Reid, and Gustavo Carneiro. Training medical image analysis systems like radiologists. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 546–554. Springer, 2018.

Brian McFee and Gert R Lanckriet. Metric learning to rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 775–782, 2010.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Computer Vision–ECCV 2012*, pages 488–501. Springer, 2012.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. *arXiv preprint arXiv:1707.03141*, 2017.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. 2018.

Tom M Mitchell and Sebastian B Thrun. Explanation-based neural network learning for robot control. In *Advances in neural information processing systems*, pages 287–294, 1993.

Tom M Mitchell et al. Machine learning, 1997.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563, 2017.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3661–3670, 2018.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

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

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Cikm*, volume 5, page 3, 2000.

Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 151–159, 2020.

Boris N Oreshkin, Alexandre Lacoste, and Pau Rodriguez. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

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

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017.

Pedro O Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8004–8013, 2018.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. *arXiv preprint arXiv:1909.04630*, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

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

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

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

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL http://www.idsia.ch/~juergen/diploma.html.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2): 227–244, 2000.

Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011.

Daniel L Silver, Ryan Poirier, and Duane Currie. Inductive transfer with context-sensitive neural networks. *Machine Learning*, 73(3):313, 2008.

Daniel L Silver, Geoffrey Mason, and Lubna Eljabu. Consolidation using sweep task rehearsal: overcoming the stability-plasticity problem. In *Canadian Conference on Artificial Intelligence*, pages 307–322. Springer, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017a.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017b. URL http://arxiv.org/abs/1703.05175.

Shuhan Tan, Xingchao Peng, and Kate Saenko. Generalized domain adaptation with covariate and label shift co-alignment. *arXiv preprint arXiv:1910.10320*, 2019.

Antonio Torralba, Alexei A Efros, et al. Unbiased look at dataset bias. In *CVPR*, volume 1, page 7. Citeseer, 2011.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

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

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984.

Vladimir Vapnik. 24 transductive inference and semi-supervised learning. 2006.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

Geoffrey I Webb and Kai Ming Ting. On the application of roc analysis to predict classification performance under varying class distributions. *Machine learning*, 58 (1):25–32, 2005.

Jun Wen, Nenggan Zheng, Junsong Yuan, Zhefeng Gong, and Changyou Chen. Bayesian uncertainty matching for unsupervised domain adaptation. *arXiv preprint arXiv:1906.09693*, 2019.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. *arXiv preprint arXiv:1903.01689*, 2019.

Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 5419–5428, 2018.

Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 433–443, 2019a.

Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809, 2018.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413, Long Beach, California, USA, 09–15 Jun 2019b. PMLR.

Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep meta-learning: Learning to learn in the concept space. *arXiv preprint arXiv:1802.03596*, 2018.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2020.

Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.

Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.