Advancing Multi-Domain Active Learning

Strategies and Techniques Using Neural Networks for Classification

by

Rui He

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
July 2023

ABSTRACT

Dealing with data collected from various domains has emerged as a pervasive challenge in machine learning applications, necessitating the use of multi-domain learning (MDL) to efficiently process and learn from the collected multi-domain data, which have different distributions. Nevertheless, constructing high-quality annotated multi-domain datasets can be both challenging and expensive due to the difficulties in accessing multiple domain experts. Active learning (AL) provides a potential solution to this issue, effectively reducing labeling costs by only selecting and acquiring labels for the most informative data instances. However, conventional AL methods are not designed to handle multi-domain data and cannot be directly applied to multiple domains. Moreover, there is a notable scarcity of studies exploring the intersection of MDL and AL. To fill this gap, the focus of this thesis is on multi-domain active learning (MDAL), an innovative approach that combines active learning and multi-domain learning, enabling cost-effective learning through the use of neural networks.

In this thesis, we ***first*** conduct a comprehensive review of AL and MDL research, formalizing the problem definitions and settings for MDAL. An open-source awesome active learning knowledge base is also developed to facilitate our research and the community. ***Second***, to address the problem of MDAL, a unified pipeline is proposed to integrate conventional AL methods with neural network-based multi-domain learning models. Extensive comparative experiments reveal that certain model-strategy pairs yield strong performance empirically across domains. ***Third***, to address model limitations under insufficient annotations, the thesis introduces a plug-and-play multi-domain contrastive learning method applicable to various MDL models under share-private architecture. Experimental results demonstrate significant improvement in model performance with insufficient annotations. ***Fourth***, limitations of conventional AL methods in assessing domain-shared information

motivate the development of tailored MDAL algorithms. From the labeling strategy perspective, a novel perturbation-based two-stage AL strategy is proposed to select informative cross-domain instances. This approach outperforms conventional methods by explicitly evaluating domain-shared information.

Through formalizing the problem, extensive comparative analysis, and introducing new learning methods and strategies, this thesis advances the state-of-the-art in cost-effective MDAL using deep neural networks. The proposed techniques and findings provide valuable insights to facilitate future multi-domain learning applications with limited labeled data.

Key Words: *Multi-Domain Learning; Active Learning; Multi-Domain Active Learning, Insufficient Annotations; Share-Private Architecture; Contrastive Learning; Perturbation*

ACKNOWLEDGMENTS

As a student of the Joint PhD Training Program offered by the University of Birmingham (UoB) and Southern University of Science and Technology (SUSTech), I would like to express my heartfelt gratitude to my supervisors Dr. Shan He in UoB and Prof. Ke Tang in SUSTech for their unwavering support throughout my academic journey. Their professional guidance, encouragement, and kindness not only enriched my research skills but also shaped me as an individual. Without their invaluable advice, this thesis would not have been possible. I would also like to thank UoB and SUSTech for their support for my studies in both universities.

Many thanks to Dr. Yu Zhang for his valuable discussions, suggestions, and encouragements. I am truly appreciated for his friendship.

I would like to thank my thesis group members, Prof. Ata Kaban, Dr. Hyung Jin Chang for their constructive feedback and insightful suggestions during my PhD study. I also want to acknowledge the early guidance provided by Prof. Hu Xu, Prof. Haizhou Lu, my undergraduate supervisors, which laid a strong foundation for my research career. Additionally, I would like to thank Dr. Peter Hancox for providing valuable research skill training. I would also like to thank many administrative staffs, Ms. Sarah Brookes and Ms. Kate Sterne (UoB student administrators), Mr. Xuefeng Zhang and Mr. Xing Zhang (SUSTech student administrators), Ms. Juan Tang (NICAL Lab administrator), Dr. Hong Wen (SUSTech clinic doctor).

I am fortunate to have had the support and friendship of numerous friends and colleagues during my studies at UoB and SUSTech. I want to express my gratitude to Dr. Chengbin Hou, Dr. Rujia Li, Dr. Wei Chen, Dr. David McDonald, Mingyang Feng from UoB, and Dr. Shengcai Liu, Dr. Wenjie Chen, Dr. Weijie Zheng, Dr. Bo Yuan, Dr. Peng Yang, Dr. Xiaofeng Lu, Dr. Wenjing Hong, Dr. Guiyin Li, Zhiyuan Wang, Zeyu Dai, Jiahao Wu, Ning Lu, Xuanfeng Li, Hui Ouyang, Shaofeng Zhang, Qixin Guo, Qi Yang, Muyao Zhong, Fu Peng, Lan Tang, Dr. Feiyu E, Dr. Mingzhe Li, Dr. Chenguang Liu, Dr. Zezheng Feng,

Jianjun Chen, Weihuang Wen, Dr. Yi Liu from SUSTech. Your friendship and mentorship have been invaluable to my growth and learning.

Throughout my PhD journey, I have also been fortunate to receive encouragement and support from many old friends across the world: Dr. Yan Han, Bing Qiao, Qi Xu, Chenqi Liang, Jian Zhang, Ruixia Sun, Wenyan Zhong, Yuchen Fu, Xinyu Xi, Xiangpei Kong, Yingnian Qiao, Xiaoran Xiang. Special thanks go to my girlfriend, Yuechen Wei, for her support and encouragement during the most challenging periods of my Ph.D. studies.

Lastly, I extend my deepest gratitude to my parents and my family for their love and support. My gratitude is beyond words.

In loving memory, I dedicate this doctoral dissertation to my grandparents, whose love and expectations will forever be cherished. May they rest in peace.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

$\boldsymbol{x}$ input vector.

y category or label for classification tasks.

σ activation function.

$\hat{y}$ predicted output.

L loss function.

$\boldsymbol{\theta}$ parameters of the model.

$\hat{E}(\boldsymbol{\theta})$ empirical error over the sample dataset D.

$\tilde{E}(\boldsymbol{\theta})$ structual error including the regularization term.

$\Omega(\cdot)$ regularization term.

η learning rate.

$sim(x_i, x_j)$ similarity between embeddings x_i and x_j.

$\mathbb{E}$ expectation or expected value.

∇ gradient operator.

$\| \cdot \|$ norm.

O big O notation for algorithmic complexity.

τ temperature parameter in contrastive learning.

$\mathcal{D}$ domain consisting of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$.

$\mathcal{T}$ task consisting of a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$.

$\mathcal{L}_\alpha$ informative labeled set selected by an AL acquisition function α.

B overall labeling budget for active learning.

$\mathcal{U}$ unlabeled data set.

$\mathcal{P}$ data pools containing all the available data.

List of Acronyms

AL active learning.

CNN convolutional neural network.

CV computer vision.

DA domain adaptation.

DNN deep neural network.

EM expectation-maximisation.

ERM empirical risk minimisation.

GAN generative adversarial networks.

MDAL multi-domain active learning.

MDL multi-domain learning.

MLP multilayer perceptron.

MSE mean squared error.

MTL multi-task learning.

NLP natural language processing.

ReLU rectified linear unit.

SGD stochastic gradient descent.

SVMs support vector machines.

Chapter 1

Introduction

Handling data collected from various domains has become a pervasive challenge in machine learning applications, including computer vision (CV) (Krizhevsky et al., 2012), natural language processing (NLP) (Sutskever et al., 2014), and speech recognition (Hinton, Deng, et al., 2012). Domain usually refers to a probabilistic distribution, which is a set of data instances that share the same characteristics. In real applications, medical images from different scanners, paintings with different styles, reviews under different products could be considered as different domains. The inclusion of multi-domain data enhances both the volume and diversity of the dataset, which is advantageous for training models with enhanced generalization capabilities. This rises a requirement for multi-domain learning (MDL) (Dredze and Crammer, 2008), which aims to efficiently process and learn from data originating from multiple domains and distributions. However, the effort required for annotations remains a significant hurdle in this multi-domain data scenario, necessitating cost-effective multi-domain learning solutions, particularly in the era of data-driven deep learning. In this study, we focus on applying active learning (Settles, 2009) to multi-domain data scenarios, specifically termed as multi-domain active learning (MDAL) (Li, Jin, et al., 2012), aiming to alleviate the burden of labeling by selecting the most informative instances for annotation. We approach this from both model training and active labeling strategy perspectives to facilitate cost-effective

multi-domain active learning.

In this chapter, we provide an introduction to the general background, scope, research questions, and contributions of this thesis. The subsequent sections are structured as follows: In Section 1.1, a brief introduction is provided on the concept of multi-domain active learning, along with the motivation behind this thesis. Section 1.2 addresses any potential ambiguities concerning the scope of this thesis. Following that, Section 1.3 outlines the research questions that will be addressed. Section 1.4 highlights the contributions made by this thesis. In Section 1.5, the published and submitted papers resulting from this thesis are listed. Lastly, Section 1.6 provides an overview of the organizational structure of this thesis.

1.1 Background

Over the past several decades, the field of machine learning has experienced considerable growth and has been successfully deployed to address a diverse range of real-world challenges across various fields, encompassing computer vision (CV) (Krizhevsky et al., 2012), natural language processing (NLP) (Sutskever et al., 2014), and speech recognition (Hinton, Deng, et al., 2012), among others. The empirical machine learning methods, fueled by data, have consistently demonstrated their effectiveness and utility in plenty of applications, integral to our daily lives. The quality and volume of data play a crucial role in determining the performance and accuracy of a given model. In practical applications, the data utilized for a particular task is often collected from a variety of sources, resulting in a scenario known as multi-domain data. Data from different domains vary in distribution but share certain common characteristics (Pan and Yang, 2010). For instance, in computer vision, multi-domain data could comprise information gathered from various cameras, sensors, or even differing seasonal conditions. Similarly, in the field of natural language processing, data collected from diverse news websites, disparate social media platforms, or different languages would constitute multi-domain data. Learning from such multi-domain data could be essential in enhancing both the performance and the generalization capacity of the model. Multi-domain

learning (MDL) (Dredze and Crammer, 2008) constructs models that can efficiently process and learn from data originating from multiple domains, thereby maximizing their accuracy and generalization potential.

Although multi-domain learning effectively utilizes data from multiple domains, the multi-domain data collection process can be expensive, making the accumulation of sufficient data for model training challenging. Compared to single-domain learning, the issue of labeling cost is exacerbated in MDL, as constructing a labeled multi-domain dataset is more complex due to the difficulty in accessing data from multiple domain experts (Huang, Han, et al., 2019; Zheng et al., 2021; Mghabbar and Ratnamogan, 2020). The expense of labeling multi-domain data is not only financial but also temporal, computational, and cognitive, as it demands significant time investment from domain experts, substantial computational resources, and places a high cognitive burden on experts. For instance, in the case of multi-domain medical image datasets (Huang, Han, et al., 2019), the high financial cost of accurate annotations from medical experts across various research fields is just one of the challenges. Cognitively, the complexity of understanding and accurately annotating data across varied domains places a high cognitive burden on experts. Added to this are the differing privacy and legal concerns, quality assurance processes, and labeling tools across domains, all of which increase costs. As a result, labeling data from multiple domains is significantly more expensive than from a single domain, and the high cost of annotation often results in a lack of labeled data, negatively impacting model performance and generalization ability.

Existing studies on multi-domain learning primarily focus on constructing models that can efficiently utilize shared information across multiple domains, while they have not adequately addressed the cost issue (Liu, Qiu, et al., 2017). The high labeling cost is also prevalent in conventional single-domain learning, which has been extensively studied in the literature. Active learning (AL) (Settles, 2009) is a popular solution for this issue. Given an annotation budget, an active learning approach could build a labeled dataset from scratch by selecting the most informative instances to annotate, thereby improving model performance at a lower annotation cost. Empirical evidence (Zhan, Liu, et al., 2021) suggests that active

learning is effective in reducing annotation costs, making it a promising solution for high labeling cost issues in multi-domain learning.

Multi-domain active learning (MDAL), which combines multi-domain learning and active learning, was first defined by Li, Jin, et al. (2012). The MDAL approach aims to select the most informative instances from multiple domains for annotation, providing a promising solution to the high labeling cost issue in multi-domain learning. To date, only a handful of studies have been conducted on MDAL, and these are inherently limited as they are tailored for ad hoc tasks on specific types of conventional models, such as support vector machines (SVMs) (Li, Jin, et al., 2012) and Rating-Matrix Generative Model (Zhang, Jin, et al., 2016). Therefore, the performance of active learning in multi-domain learning on more general tasks with advanced neural networks remains unclear.

Motivated by these issues and previous works, **this thesis focuses on the multi-domain learning scenario with fewer labeled instances and proposes cost-effective solutions by using active learning with neural networks.** Specifically, we first explore the potential of multi-domain active learning with existing adaptable methods within a unified MDAL pipeline. Based on this pipeline, we then analyze and enhance the performance of MDAL from both model training and active labeling strategy perspectives.

1.2 Scope of the Thesis

This thesis exclusively focuses on the multi-domain learning setting, which involves **simultaneously** learning the **same task** across **multiple distinct domains** within a **same input space**, with **performance evaluation conducted across all domains**. The experiments and analysis presented in this thesis primarily utilize neural-network-based models (LeCun, Bengio, et al., 2015). The specific definitions and details of the corresponding settings will be provided in Chapter 2.

It is important to note that the terms "multi-" and "cross-domain" are commonly used in a broader context within the field of machine learning. Many other fields in the literature

extensively reference these terms, such as domain adaptation (DA) (Pan and Yang, 2010), multi-task learning (MTL) (Zhang and Yang, 2021), multi-modal learning (Li, Yang, and Zhang, 2019), and meta learning (Hospedales et al., 2022). Although these paradigms can be considered as learning from multiple domains to some extent, they differ in terms of the data types and the specific requirements of downstream applications. While these fields may share some technical insights, they tackle distinct problem settings with different constraints.

1.3 Research Questions

As previously discussed, there is a limited amount of research conducted on multi-domain active learning (MDAL), and the existing studies are often constrained by task-specific conventional models. Consequently, the performance of active learning (AL) in the context of multi-domain learning (MDL) on more general tasks with advanced neural networks remains unclear. In this thesis, we aim to address this gap and explore the following four main research questions.

1.3.1 RQ 0: How to build a comprehensive categorization for current active learning research field?

Active learning (Settles, 2009) is a well-researched area within machine learning, and it has thoroughly demonstrated efficacy in lowering annotation costs. AL is also one of the main focus of this thesis. Despite decades of extensive research, this field remains vibrant with ongoing studies. At the mean time, the most recent literature surveys and reviews on AL (Zhan, Liu, et al., 2021; Zhan, Wang, et al., 2022), primarily concentrate on the query strategy perspective, providing only a partial picture of the field. The categorization from these surveys is not comprehensive enough to cover the entire active learning research field, specifically, for strategies, considerations, applications, etc. Consequently, we are interested in providing a comprehensive categorization. This topic will be covered in Section 2.4 as a

part of thesis background in Chapter 2.

1.3.2 RQ 1: How do conventional active learning methods perform in multi-domain active learning?

Neural networks are prevalently employed across diverse machine learning tasks, exhibiting effectiveness in numerous domains. In multi-domain learning, neural networks underpin multiple model architectures. However, no previous research has explored multi-domain active learning with neural networks, thereby limiting MDAL's full potential. Hence, we integrate active learning methods with MDL and neural networks to devise a unified MDAL pipeline. In this pipeline, traditional single-domain active learning methods can be seamlessly implemented with various neural-network-based MDL models. This topic will be covered in Chapter 3.

Utilizing the proposed MDAL pipeline, a variety of strategies and models can be combined, thereby raising the following sub-questions:

- Firstly, we aim to determine if there exists a model or an information-sharing scheme that is naturally suitable for MDAL.
- Additionally, we seek to assess whether AL can yield improvements over random selection. At the same time, we are interested in identifying any specific strategy that significantly outperforms the others.
- Moreover, we are interested in investigating whether the models and strategies that exhibit strong overall performance maintain consistent effectiveness across each domain.

1.3.3 RQ 2: From the model training perspective of MDAL, how to train MDL models under insufficient annotations?

As an active learning process, the ultimate performance of MDAL is heavily dependent on both of the MDL model and AL strategy. From the model perspective, conventional

approaches prioritize the extraction of domain-shared information and the preservation of domain-private information, adhering to the shared-private framework (SP structure), which offers significant advantages over single-domain learning. Current MDL research primarily concentrates on supervised learning settings, where sufficient annotations are presumed to be available. This assumption is impractical in real-world scenarios, given the high annotation costs and the general scarcity of labeled multi-domain data. Therefore, we question if MDL performance could be improved with insufficient annotations. This topic will be examined in Chapter 4.

We propose a novel method called multi-domain contrastive learning (MDCL) (He, Liu, Wu, et al., 2023), which is a plug-and-play method that can be applied to various models under the renowned share-private architecture (Bousmalis et al., 2016). With the proposed method, the following sub-questions arise:

- As a plug-and-play method, can MDCL enhance the performance of various models under the SP structure with the limited number of labeled instances (around 5%-20%) or extremely few labeled instances (around 1%)?
- Since there are two technically independent components in MDCL, how does each of them affect the performance?
- Given a further labeling budget, MDAL could be utilized. Can MDCL improve the entire MDAL process with a relatively large number of unlabeled instances (5%-50%)?

1.3.4 RQ 3: From the labeling strategy perspective of MDAL, how to design effective AL strategies tailored explicitly for multi-domain data?

Active learning selection strategies serve a pivotal function in Multi-Domain Active Learning (MDAL). As we have previously introduced (He, Liu, He, et al., 2023), traditional Active Learning methods can be applied to Multi-Domain Learning models as a direct solution.

However, the performance of the MDAL framework is still limited by these conventional methods, as they fail to thoroughly evaluate the domain-shared information for each instance, leading to sub-optimal active learning selection. Thus, a pressing question arises: can the MDAL performance be further augmented by devising a more suitable labeling strategy? This topic will be addressed in Chapter 5.

We propose a novel MDAL strategy called Perturbation-Based Two-Stage MDAL (P2S-MDAL) (He, Dai, et al., 2023). This method evaluates the domain-shared information of each instance by introducing perturbations to the instance and evaluating the resulting changes. With this proposed method, the following sub-questions emerge:

- As the first dedicated AL strategy for MDAL, does P2S-MDAL outperform conventional AL strategies?
- P2S-MDAL consists of two stages, to ensure the in-domain diversity and cross-domain informativeness. Whether both stages provide positive effects?
- In comparison to other conventional AL strategies, how does the time complexity of P2S-MDAL compare against the others?

1.4 Contributions

The main contributions of this thesis can be briefly summarized as follows:

- We conduct a comprehensive literature review on multi-domain active learning (MDAL) in Chapter 2. This includes providing a formal definition for MDAL and summarizing the related fields, such as multi-domain learning, active learning, and techniques for sharing domain information, notwithstanding the existing research gap in MDAL.
- We propose and maintain a well-structured open-source active learning knowledge library, named "awesome active learning", discussed in Section 2.4.2. This repository encompasses a wide range of active learning resources such as research papers, code,

books, blogs, among others. Updated monthly, this library has had a significant impact on the active learning research community[1].

- We explore the performance of traditional AL methods within MDL in Chapter 3. We introduce a new MDAL pipeline that seamlessly integrates conventional AL methods with MDL models. The first comprehensive comparative study is then conducted on this MDAL pipeline across six datasets, six models, and five AL strategies. The models and strategies that perform well empirically are recommended for MDAL, and their advantages are analyzed.

- From the model perspective, we propose an innovative method multi-domain contrastive learning (MDCL) for enhancing MDL performance with insufficient annotations in Chapter 4. MDCL is readily compatible with many renowned share-private models, requiring no additional model parameters and allowing for end-to-end training. Experimental results reveal that MDCL significantly improves performance across five textual and image multi-domain datasets. Moreover, MDCL can be employed in MDAL to achieve a superior initialization and consequently result a better overall performance.

- From the active labeling strategy perspective, we propose the first ad hoc AL strategy for improving MDAL performance in Chapter 5. We introduce a novel AL strategy named perturbation-based two-stage multi-domain active learning (P2S-MDAL), the first strategy designed specifically for the MDL scenario based on the renowned ASP-MTL model. Experimental results show top-tier performance across multiple datasets. Besides, the perturbation-based evaluation offers a fresh viewpoint in assessing the cross-domain potential of individual instances.

- We provide sufficient reading materials of MDAL such as motivation, literature reviews, concepts, representative approaches, and future directions (Chapter 1, 2, and 6).

[1]By 22 June 2023, it has been starred more than 450 times according to Github.

1.5 Publications Resulting from the Thesis

The works resulting from this thesis have been presented in the following published or submitted papers.

[1] **Rui He**, Shengcai Liu, Shan He and Ke Tang. "Multi-Domain Active Learning: Literature Review and Comparative Study". In: *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 791–804, 2023. (He, Liu, He, et al., 2023)

- Chapter 2 is partially based on this paper.
- Chapter 3 is based on this paper.

[2] **Rui He**, Shengcai Liu, Jiahao Wu, Shan He and Ke Tang. "Multi-Domain Learning From Insufficient Annotations" arXiv preprint arXiv:2305.02757 (2023). (*ECAI-2023*) (He, Liu, Wu, et al., 2023)

- Chapter 4 is based on this paper.

[3] **Rui He**, Zeyu Dai, Shan He and Ke Tang. "Perturbation-Based Two-Stage Multi-Domain Active Learning" arXiv preprint arXiv:2306.10700 (2023). (Submitted to *CIKM-2023*) (He, Dai, et al., 2023)

- Chapter 5 is based on this paper.

I have been fortunate enough to collaborate with numerous researchers on other projects. While these works extend beyond the scope of this thesis, the publications resulting from those collaborations can offer valuable insights into the future research directions of MDAL.

[4] Ning Lu, Shengcai Liu, **Rui He** and Ke Tang. "Large Language Models can be Guided to Evade AI-Generated Text Detection" arXiv preprint arXiv:2305.10847 (2023). (Submitted to *NeuraIPS-2023*) (Lu et al., 2023)

- Chapter 6 discusses several promising directions starting from this paper.

[5] Jiahao Wu, Wenqi Fan, **Rui He**, Shengcai Liu, Qing Li, Ke Tang. "Dataset Condensation for Recommendation" (To submit to *ICDE-2023*)

- Chapter 6 discusses several promising directions starting from this paper.

1.6 Outline of the Thesis

The outline of this thesis is illustrated in Figure 1.1. In Chapter 1, we have introduced the background of MDAL and summarize the research questions and contributions of this thesis. For the remaining chapters, Chapter 2 provides a formal definition and a comprehensive literature review for MDAL. Chapter 3, 4, and 5 are the main chapters of this thesis, which answer the research questions 1-3 respectively in great details. In Chapter 6, we summarize the thesis and discuss the future works.

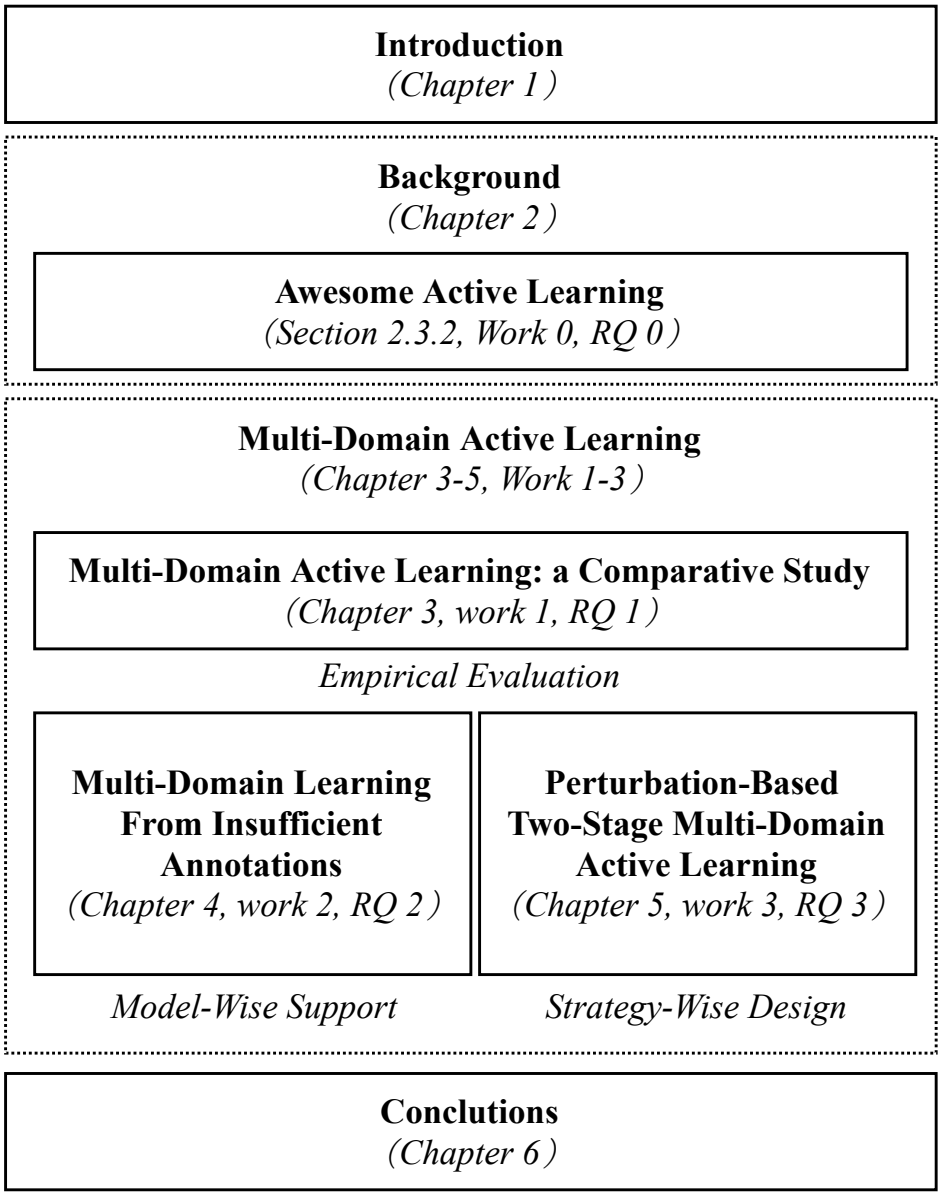


Figure 1.1: The outline of this thesis.

Chapter 2

Background

In this chapter, we introduce general background knowledge of this thesis. Initially, We will provide a brief overview of neural networks in Section 2.1. Then, Section 2.2 aims to clarify the relationships among various research fields found in the literature that often cause confusion. This is primarily due to the abundance of terms beginning with "multi-" or containing "domain". Following this, Section 2.3 presents the problem formulations of active learning, multi-domain learning, and multi-domain active learning, respectively. The subsequent sections provide an extensive literature review of closely related fields of our thesis. In Section 2.4, we review the field of active learning, which forms one of the foundational elements of our thesis. Moving forward, Section 2.5 delves into learning from multiple domains, which serves as the other foundation our thesis. Section 2.6 explores the field of annotation from multiple domains, as it closely aligns with the MDAL (multi-domain active learning) in our thesis. Finally, a summary of this chapter is presented in Section 2.7.

2.1 Neural Networks

2.1.1 Feedforward Neural Networks

Feedforward neural networks, also known as multilayer perceptrons (MLPs), are the simplest type of artificial neural network architecture (LeCun, Bengio, et al., 2015). They consist of multiple layers of nodes, each layer fully connected to the next one. The model starts from the input nodes, through the hidden layers, and finally to the output layer. The primary function of a feedforward network is to approximate some function f. For example, for a classifier, $y = f(\boldsymbol{x})$ maps an input $\boldsymbol{x}$ to a category y.

The operation within a neuron involves the weighted sum of its inputs, followed by the application of an activation function, which can be described by the following equations:

$$\begin{aligned} z_i^{(l)} &= \sum_j w_{ij}^{(l)} x_j + b_i^{(l)} \\ a_i^{(l)} &= \sigma(z_i^{(l)}) \end{aligned} \tag{2.1}$$

where x_j are the inputs to the neuron, $\boldsymbol{w}_{ij}^{(l)}$ are the weights connecting the j-th input to the i-th neuron in the l-th layer, $b_i^{(l)}$ is the bias term for the i-th neuron in the l-th layer, $z_i^{(l)}$ is the weighted sum of inputs, $a_i^{(l)}$ is the output of the neuron after applying the activation function σ, which introduces non-linearity into the model allowing it to learn complex functions. Common choices for σ include the sigmoid function, hyperbolic tangent function (tanh), and the rectified linear unit (ReLU, Agarap, 2018). Thus, the l-th layer computation could be written as:

$$f^{(l)} = \sigma(\boldsymbol{W}^{(l)}\boldsymbol{x} + \boldsymbol{b}^{(l)}) \tag{2.2}$$

The whole neural network (d layers) can be represented as a composite function

$$f(\boldsymbol{x}) = f^{(d)} \circ f^{(d-1)} \circ \cdots \circ f^{(1)}(\boldsymbol{x}) \tag{2.3}$$

The process of computing the output of a neural network by applying the input data to the input layer, then propagating it through each subsequent layer using the above equations, is

known as **forward propagation**. The final output provides the prediction of the network.

2.1.2 Optimization

Optimization in the context of neural networks refers to the process of adjusting the model's parameters (weights and biases) to minimize the discrepancy between the predicted outputs $\hat{y}$ and the true outputs y in the training data. This discrepancy is quantified by a loss function $L(y, \hat{y})$, also known as a cost function or error function.

The choice of the loss function depends on the type of task (e.g., regression, classification). For regression tasks, the mean squared error (MSE) is commonly used:

$$L_{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.4}$$

For binary classification, the binary cross-entropy (BCE) is prevalent:

$$L_{BCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{2.5}$$

And for multi-class classification, the cross-entropy is often used:

$$L_{CE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log(\hat{y}_{ic}) \tag{2.6}$$

where N is the number of samples, C is the number of classes, y_{ic} is a binary indicator of whether class c is the correct classification for observation i, and $\hat{y}_{ic}$ is the predicted probability that observation i is of class c.

With the aforementioned loss functions, the objective of a machine learning model is to minimize its generalization error, represented as:

$$E(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L(f(\boldsymbol{x}; \boldsymbol{\theta}), y)],$$

Typically, the true distribution $\mathcal{D}$ of data is unknown, and we only have a sample dataset D, presumed to be independently and identically distributed (i.i.d.) from $\mathcal{D}$. A practical

approach to approximate this is by minimizing the expected loss over the available training dataset:

$$\hat{E}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i}^{n} L\big(f\big(\boldsymbol{x}^{(i)};\boldsymbol{\theta}\big), y^{(i)}\big).$$

This strategy is known as empirical risk minimization (ERM). Nevertheless, optimizing a model on a limited training dataset often leads to overfitting, which is the model's inability to generalize well to unseen data. To mitigate this, it's essential to find a balance between the model's complexity and its performance on the training data, a concept known as structural risk minimization (SRM, Sain, 1996). There are several methods to apply the SRM principle, often referred to as regularization. A common technique involves adding a penalty term $\Omega(\boldsymbol{\theta})$ to the loss function, adjusting the optimization goal to:

$$\tilde{E}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i}^{n} L\big(f\big(\boldsymbol{x}^{(i)};\boldsymbol{\theta}\big), y^{(i)}\big) + \lambda\Omega(\boldsymbol{\theta}),$$

where $\lambda \in [0, \infty)$ is a hyperparameter that tunes the significance of the regularization term. A typical choice for the Ω is the L^2 norm, $\Omega(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_2^2$, which is also known as weight decay. Alternatively, the L^1 norm, $\Omega(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = \sum_i |\theta_i|$ can be used. L^1 regularization encourages sparsity in the parameter vector $\boldsymbol{\theta}$, effectively promoting feature selection by driving more $\boldsymbol{\theta}_i$ values to zero compared to L^2 regularization.

To minimize the structural risk, neural networks commonly use gradient descent or its variants. **Gradient descent** (Cauchy et al., 1847) is an iterative optimization algorithm that adjusts parameters in the direction of the negative gradient of the loss function with respect to the parameters. The parameters are updated as follows at time step t:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla E(\boldsymbol{\theta}_t) \tag{2.7}$$

where $\boldsymbol{\theta}$ represents the parameters of the model, η is the learning rate, and $\nabla E(\boldsymbol{\theta})$ is the

gradient of the loss function with respect to the parameters. Backpropagation is the cornerstone algorithm for training neural networks, allowing the gradient of the loss function to be efficiently computed for each layer in the network. This algorithm uses the chain rule of calculus to compute gradients of the loss with respect to each parameter by moving backwards through the network, from the output layer to the input layer.

2.1.3 Contrastive Learning and Data Augmentation

Contrastive learning is a technique in machine learning that learns representations by contrasting positive pairs against negative pairs. This approach has gained significant traction in the field of unsupervised learning, particularly for tasks involving images, text, and audio. In the context of machine learning, contrastive learning aims to learn effective representations without the need for explicit labels, leveraging the structure inherent in the data. This approach is especially powerful in domains where obtaining labeled data is expensive or infeasible.

One popular loss function in contrastive learning is the NT-Xent contrastive loss (Chen, Kornblith, et al., 2020), also called InfoNCE loss, which has been widely used in self-supervised learning frameworks. It is defined as:

$$L_{InfoNCE} = -\log \frac{\exp(sim(x_i, x_j)/\tau)}{\sum_{k=1}^{N} \exp(sim(x_i, x_k)/\tau)} \tag{2.8}$$

where $sim(x_i, x_j)$ denotes the similarity between embeddings x_i and x_j, N is the number of negative samples, and τ is a temperature parameter that controls the separation of the distributions.

Data augmentation (Shorten and Khoshgoftaar, 2019a) plays a crucial role in contrastive learning by generating diverse positive pairs, which helps the model learn more robust and invariant features. Common data augmentation techniques include cropping, flipping, rotating images, adding noise, or changing the color balance for visual data. For text, augmentations may involve synonym replacement, sentence shuffling, or back-translation (Feng, Gangal,

et al., 2021). By effectively leveraging the relationships between data points and generating varied augmentations, models can learn meaningful and generalizable features.

2.2 Relations among Different Research Fields

In this section, we aim to clarify the relationships among various research fields that are frequently confusing in the literature. This confusion primarily arises from the abundant use of terms starting with "multi-" or containing "domain". For example, there is often confusion between multi-domain learning (MDL) and multi-task learning (MTL) (Zhang and Yang, 2021), because both approaches seek to acquire shared knowledge from multiple individual tasks. Additionally, multi-domain learning and domain adaptation (DA) (Pan and Yang, 2010) are frequently confused due to the existence of multiple domains. Consequently, it is essential to provide a clear understanding of the relationships among these fields.

Before we proceed, we need to clarify the definitions of the terms "domain" and "task" in the context of machine learning. Previous research (Pan and Yang, 2010) has provided clear definitions and examples for these terms, which we adopt in this thesis.

Definition 1 (Domain) *A domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in X$.*

Definition 2 (Task) *Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ (denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which is not observed but can be learned from the training data, which consist of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be used to predict the corresponding label, $f(x)$, of a new instance x. From a probabilistic viewpoint, $f(x)$ can be written as $P(y \mid x)$.*

For example, if our learning task is document classification, and each term is taken as a binary feature, then $\mathcal{X}$ is the space of all term vectors, x_i is the i^{th} term vector corresponding to some documents, and X is a particular learning sample. In general, if two domains are

different, then they may have different feature spaces (dimensions of features) or different marginal probability distributions (different languages, topics or qualities). Besides, in our document classification example, $\mathcal{Y}$ is the set of all labels, which is True, False for a binary classification task. If two tasks are different, they may have different label spaces (dimensions of outputs) or objective predictive function (document sentiments or authenticity).

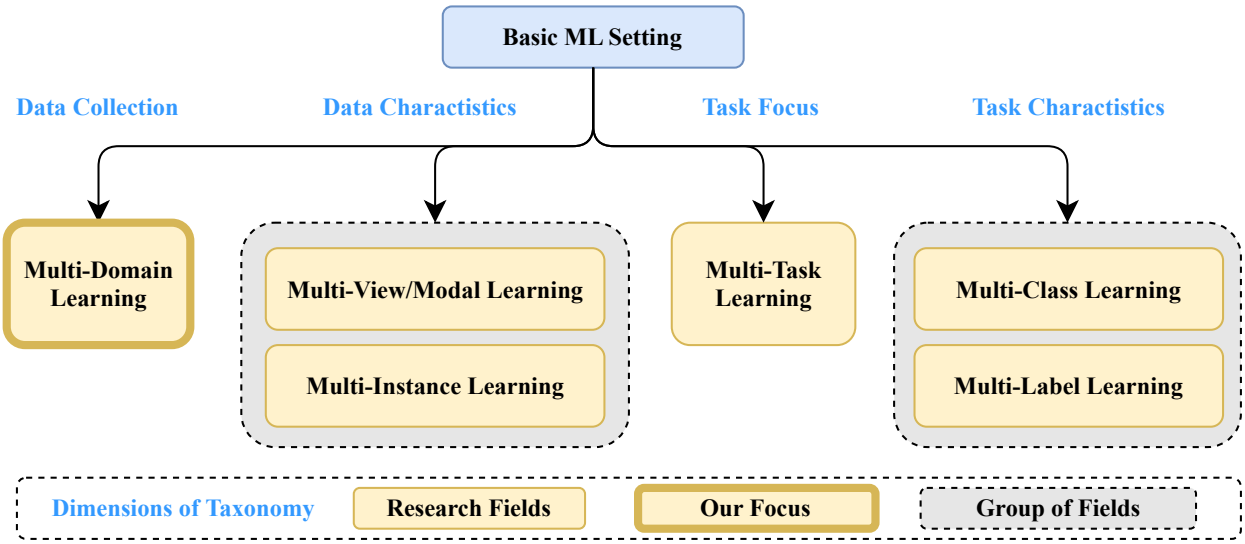


Figure 2.1: The relations of different research fields termed with "multi-".

Then, we will briefly clarify the relationships among various "multi-" fields, as depicted in Figure 2.1. We begin with a basic Machine Learning scenario in which data is collected from a single domain $\mathcal{D}$ and consists of a single view/modal, aiming to perform a binary classification task with a single label. This simplest setting can be expanded along four dimensions: data collections, data characteristics, task focus, and task characteristics.

1. **Data Collections:** When the data collection is from multiple distributions $\mathcal{D}$, it falls under the category of multi-domain learning.

2. **Data Characteristics:** In cases where the feature space $\mathcal{X}$ of the collected data exhibits unusual characteristics, the learning mission can be classified as multi-view/modal learning (Sa, 1993) or multi-instance learning (Dietterich et al., 1997). Specifically, in multi-view/modal learning, each instance is described in more than two certain views

or modalities at the same time. Besides, in multi-instance learning, each instance is a bag of sub-instances, and the label is assigned to the whole bag.

3. **Task Focus:** When there are multiple objective predictive functions $f(\cdot)$ for a certain domain, the learning mission evolves into multi-task learning.

4. **Task Characteristics:** In the case where the label space $\mathcal{Y}$ of the task exhibits unusual characteristics, i.e. larger label space, the learning mission can be classified as multi-label learning (Zhang and Zhou, 2014) and multi-class learning.

It is worth noting that, these dimensions are not mutually exclusive and can be combined to form more complex learning settings.

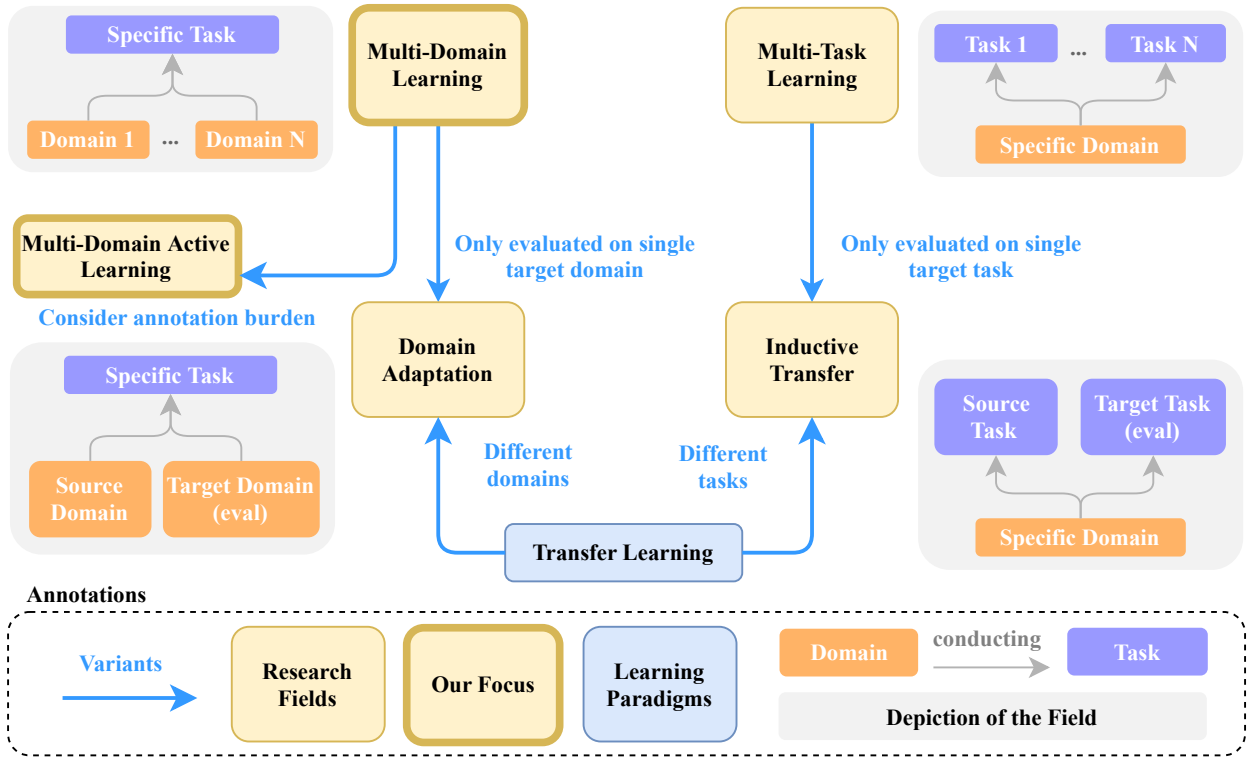


Figure 2.2: The field map of multi-domain learning, multi-task learning, domain adaptation and transfer learning.

In this thesis, we focus on the vanilla multi-domain learning scenario, whose formal definition will be provided in Section 2.3.2. Before that, since the terminologies sometimes

are misused in the literature, we will further clarify the relationships among multi-domain learning, multi-task learning, domain adaptation (DA) and transfer learning in Figure 2.2. MDL is typically focusing on the same task across multiple domains. Conversely, MTL centers on different tasks within the same domain. DA, on the other hand, falls under the paradigm of transfer learning (Pan and Yang, 2010), where knowledge is transferred from one domain to another. DA solely emphasizes the performance of the target domain, while MDAL focuses on the performance across all domains.

For instance, in the context of annotated shopping review data, the reviews pertaining to a specific product can be regarded as a single domain. These reviews can then be utilized to train models for various tasks such as sentiment classification and named entity recognition (Nadeau and Sekine, 2007). MTL enables the simultaneous construction of models for multiple tasks within one domain, specifically on the reviews of one particular product. In contrast, MDL constructs models using multiple domains, encompassing reviews from different products, but focusing only on one task. Similar to MDL, DA also concentrates on a single task. However, DA only aims to build a model for a target domain (reviews of another product) by utilizing the data from a source domain (annotated reviews of the current product).

2.3 Problem Formulations

In this section, we will provide definitions for the key terms related to this thesis: active learning, multi-domain learning, and multi-domain active learning, respectively.

2.3.1 Active Learning

Definition 3 (Active learning) *AL (Settles, 2009) is to reduce the labeling cost by only annotating informative instances rather than all the unlabeled data. The informative labeled set* $\mathcal{L}_\alpha = \{(x_i, y_i)\}_{i=1}^{B}$ *is selected from the unlabeled data* $\mathcal{U}$*. The selection process is guided by an AL acquisition function (strategy)* α*, which can be a ranking function or a sampling*

function depends on the corresponding AL method. B is the number of labeled instances to be selected, which is the labeling budget. In the meantime, a model $f_{\theta^*(\mathcal{L}_\alpha)}$ can be trained by using the selected labeled data $\mathcal{L}_\alpha$ by the following objective:

$$\theta^*(\mathcal{L}_\alpha) = \arg\min_\theta \frac{1}{B} \sum_{i=1}^{B} L(f_\theta(x_i), y_i) \tag{2.9}$$

where L is the loss function that measures the discrepancy between the predictions of the model f_θ and the ground truth labels. The unlabeled data $\mathcal{U}$ can be pre-collected as a data pool (pool-based AL) or come in a steam manner (steam-based AL).

The utilization of pre-collected data pools is a prevalent practice, leading to extensive research on pool-based active learning (AL) in the existing literature[1]. In alignment with this data pool setting, our thesis exclusively concentrates on pool-based AL. The primary objective of pool-based AL is to maximize the utility of the model while minimizing the cost associated with labeling. This objective can be formulated as a bilevel optimization problem (Vicente and Calamai, 1994).

Definition 4 (Pool-based active learning as bilevel optimization) *Pool-based AL is to select most informative instances $\mathcal{L}$ from the unlabeled data pool $\mathcal{U}$ pre-obtained from the underlying distribution $\mathcal{D}$. The learning objective can be written as a bilevel optimization problem:*

$$\begin{aligned} &\textit{Outer Problem:} \quad \min_{\mathcal{L}\subseteq\mathcal{U},|\mathcal{L}|=B} \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(f_{\theta^*(\mathcal{L})}(x), y)] \\ &\textit{Inner Problem:} \quad \theta^*(\mathcal{L}) = \arg\min_\theta \frac{1}{B} \sum_{i=1}^{B} L(f_\theta(x_i), y_i) \end{aligned} \tag{2.10}$$

where $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^B$ is the set of labeled data. f_θ is the model parameterized by θ. L is the loss function that measures the discrepancy between the predictions of the model and the ground truth labels. $\theta^*(\mathcal{L})$ is the optimal parameters learned from the labeled dataset $\mathcal{L}$. B is the number of labeled instances to be selected, which is the labeling budget.

[1] https://github.com/SupeRuier/awesome-active-learning

In the inner problem, we are learning the model parameters θ given the labeled data $\mathcal{L}$. In the outer problem, we are trying to select the most informative data points to be labeled such that the model's generalization error is minimized. Note that the optimal θ in the outer problem is a function of the labeled data $\mathcal{L}$, hence the notation $\theta^*(\mathcal{L})$.

While the definition of bilevel optimization is clear and formal, solving it is not an easy task. In practical terms, active learning is typically framed as a sequential selection process, where instances are chosen iteratively and interactively. The term "active" stems from the interactive query and selection process (Cohn et al., 1994), which can be defined as follows.

Definition 5 (Pool-based active learning as a sequencial selection) *Pool-based AL is to select informative instances from the unlabeled data pool $\mathcal{U}_0$ which has been obtained at the very beginning. The instances are iteratively selected according to an AL acquisition function α. First, a base model $f_{\theta^*(\mathcal{L}_0)}$ is trained on the initial labeled data $\mathcal{L}_0$. Then, in the i-th AL iteration, a batch of to-be-queried instances $\mathcal{Q}_i$ is selected from unlabeled data pool $\mathcal{U}_{i-1}$ according to the selection criteria α, and then annotated by an oracle:*

$$\mathcal{Q}_i = \alpha(f_{\theta^*(\mathcal{L}_{i-1})}, \mathcal{U}_{i-1}), \quad where \quad \mathcal{Q}_i \subseteq \mathcal{U}_{i-1}, |\mathcal{Q}_i| = b \tag{2.11}$$

where b is the budget for the current iteration. $\mathcal{L}_{i-1}$ and $\mathcal{U}_{i-1}$ are then updated with the selected batch $\mathcal{Q}_i$, i.e., $\mathcal{L}_i = \mathcal{L}_{i-1} \cup \mathcal{Q}_i$ and $\mathcal{U}_i = \mathcal{U}_{i-1} \backslash \mathcal{Q}_i$. In the meantime, the model $f_{\theta^*(\mathcal{L}_i)}$ is trained on the updated data $\mathcal{L}_i = \{(x_j, y_j)\}_{j=1}^{|\mathcal{L}_i|}$ with the following objective.

$$\theta^*(\mathcal{L}_i) = \arg\min_{\theta} \ \frac{1}{|\mathcal{L}_i|} \sum_{j=1}^{|\mathcal{L}_i|} L(f_\theta(x_j), y_j) \tag{2.12}$$

The labeling process terminates once the labeling budget B is exhausted or the desired performance has been reached. Finally, the labeled set $\mathcal{L}_i$ and the model $f_{\theta^*(\mathcal{L}_i)}$ at the final iteration are obtained as the outputs.

2.3.2 Multi-Domain Learning

Multi-domain learning (MDL) represents a significant area of focus in this thesis. The primary objective of MDL is to construct a model or a collection of models that can effectively leverage data from multiple domains.

Definition 6 (Multi–domain learning) *Given K different domains (distributions) $\mathcal{D} = \{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(K)}\}$, a set of data pools $\mathcal{P} = \{\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(K)}\}$ containing both labeled and unlabeled data is collected from $\mathcal{D}$ in advance. The labeled data from each pool constitute a labeled data set $\mathcal{L} = \{\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \ldots, \mathcal{L}^{(K)}\}$, where $\mathcal{L}^{(k)} = \left\{(x_i^{(k)}, y_i^{(k)})\right\}_{i=1}^{|\mathcal{L}^{(k)}|}$ for $k = 1, 2, ..., K$. MDL is to find a set of models $F_\Theta = \{f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_K}\}$ for K domains by utilizing the common knowledge of different domains, which can be expressed as follows:*

$$\Theta^*(\mathcal{L}, \mathcal{P}) = \arg\min_{\Theta} \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{|\mathcal{L}^{(k)}|} \sum_{i=1}^{|\mathcal{L}^{(k)}|} L(F_\Theta(x_i^{(k)}), y_i^{(k)}) \right) + \Omega(F_\Theta, \mathcal{P}) \tag{2.13}$$

where L is the loss function that measures the discrepancy between the predictions of the model and the ground truth labels. $\Omega(F_\Theta, \mathcal{P})$ denotes a general structural risk term on the model parameters Θ and data pools $\mathcal{P}$ for capturing the common knowledge.

The structural risk $\Omega(F_\Theta, \mathcal{P})$ term varies in its treatment across different MDL methods. Typically, the Ω term incorporates supervision signals from $\mathcal{P}$ beyond the labels. For instance, it might constitute a contrastive loss term (He, Liu, Wu, et al., 2023) or a domain confusion term (Ganin et al., 2016). Moreover, this term can be interpreted as an additional regularization aimed at capturing shared knowledge across diverse domains, playing a crucial role in MDL.

2.3.3 Multi-Domain Active Learning

Based on the definitions of MDL and AL, we can naturally establish the concept of Multi-Domain Active Learning (MDAL). Intuitively, MDAL can also be represented as a bilevel optimization problem, which is expressed as follows.

Definition 7 (Multi–domain active learning as bilevel optimization) *Given K different domains (distributions) $\mathcal{D} = \{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(K)}\}$, a set of unlabeled data pools $\mathcal{P} = \{\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(K)}\}$ is collected from $\mathcal{D}$ in advance. MDAL is to reduce the labeling cost by only selecting informative instances $\mathcal{L} = \{\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \ldots, \mathcal{L}^{(K)}\}$ from the data pool $\mathcal{P}$ in K different domains for building a multi-domain model $F_\Theta = \{f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_K}\}$. This can be expressed as the following bilevel optimization problem:*

$$\begin{aligned} &\textit{Outer Problem:} \quad \min_{\mathcal{L}\subseteq\mathcal{P}, \sum_{k=1}^{K}|\mathcal{L}^{(k)}|=B} \ \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{(x,y)\sim\mathcal{D}^{(k)}}[L(f_{\Theta^*(\mathcal{L})}(x), y)] \\ &\textit{Inner Problem:} \quad \Theta^*(\mathcal{L}, \mathcal{P}) = \arg\min_{\Theta} \ \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{|\mathcal{L}^{(k)}|}\sum_{i=1}^{|\mathcal{L}^{(k)}|} L(F_\Theta(x_i^{(k)}), y_i^{(k)})\right) + \Omega(F_\Theta, \mathcal{P}) \end{aligned} \tag{2.14}$$

where $\mathcal{L}^{(k)} = \left\{(x_i^{(k)}, y_i^{(k)})\right\}_{i=1}^{|\mathcal{L}_k|}$ for $k = 1, 2, ..., K$. $B = \sum_{i=1}^{K}|\mathcal{L}_i|$ is the number of labeled instances to be selected, which is the labeling budget. F_Θ is the multi-domain model parameterized by Θ. L is the loss function that measures the discrepancy between the predictions of the model and the ground truth labels. $\Omega(F_\Theta, \mathcal{P})$ denotes a term on the set of data pools $\mathcal{P}$ for capturing the common knowledge through F_Θ. $\Theta^*(\mathcal{L}, \mathcal{P})$ is the optimal parameters learned from the labeled data $\mathcal{L}$ and data pool $\mathcal{P}$.

Similar to single-domain active learning, solving the bilevel optimization version of MDAL directly poses significant challenges. Therefore, a practical solution requires reformulating MDAL as an iterative selection process, which can be expressed as follows:

Definition 8 (Multi–domain active learning as a sequencial selection) *Given K different domains (distributions) $\mathcal{D} = \{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(K)}\}$, a set of unlabeled data pools $\mathcal{P} = \{\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(K)}\}$ is collected from $\mathcal{D}$ in advance. The initial labeled and unlabeled data set can be written as $\mathcal{L}_0 = \{\mathcal{L}_0^{(1)}, \mathcal{L}_0^{(2)}, \ldots, \mathcal{L}_0^{(K)}\}$ and $\mathcal{U}_0 = \{\mathcal{U}_0^{(1)}, \mathcal{U}_0^{(2)}, \ldots, \mathcal{U}_0^{(K)}\}$. Besides, for a domain k, $\mathcal{P}^{(k)} = \mathcal{L}_0^{(k)} \bigcup \mathcal{U}_0^{(k)}$, and $|\mathcal{L}_0^{(k)}| \ll |\mathcal{U}_0^{(k)}|$. MDAL is to reduce the labeling cost by iteratively selecting informative instances according to an AL acquisition*

function α from the unlabeled data $\mathcal{U}_0$ in K different domains. First, a multi-domain model $F_{\Theta^*(\mathcal{L}_0,\mathcal{P})}$ is trained on the initial labeled data $\mathcal{L}_0$ and the data pool $\mathcal{P}$. Then, in the i-th AL iteration, a batch of to-be-queried instances $\mathcal{Q}_i$ is selected from unlabeled data pool $\mathcal{U}_{i-1}$ according to the selection criteria α, and then annotated by an oracle:

$$\mathcal{Q}_i = \alpha(F_{\Theta^*(\mathcal{L}_{i-1},\mathcal{P})}, \mathcal{U}_{i-1}), \quad where \quad \mathcal{Q}_i \subseteq \mathcal{U}_{i-1}, |\mathcal{Q}_i| = b \tag{2.15}$$

where b is the budget for the current iteration. $\mathcal{L}_{i-1}$ and $\mathcal{U}_{i-1}$ are then updated with the selected batch $\mathcal{Q}_i$, i.e., $\mathcal{L}_i = \mathcal{L}_{i-1} \cup \mathcal{Q}_i$ and $\mathcal{U}_i = \mathcal{U}_{i-1} \backslash \mathcal{Q}_i$. In the meantime, the model $F_{\Theta^*(\mathcal{L}_i,\mathcal{P})}$ is trained on the updated data $\mathcal{L}_i = \{\mathcal{L}_i^{(1)}, \mathcal{L}_i^{(2)}, \ldots, \mathcal{L}_i^{(K)}\}$ where $\mathcal{L}_i^{(k)} = \left\{(x_j^{(k)}, y_j^{(k)})\right\}_{j=1}^{|\mathcal{L}_j^{(k)}|}$ and data pool $\mathcal{P}$ with the following objective.

$$\Theta^*(\mathcal{L}_i, \mathcal{P}) = \arg\min_{\Theta} \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{|\mathcal{L}_i^{(k)}|} \sum_{i=1}^{|\mathcal{L}_i^{(k)}|} L(F_\Theta(x_j^{(k)}), y_j^{(k)}) \right) + \Omega(F_\Theta, \mathcal{P}) \tag{2.16}$$

The labeling process terminates once the labeling budget B is exhausted or the desired performance has been reached. Finally, the labeled set $\mathcal{L}_i$ and the model $F_{\Theta^*(\mathcal{L}_i,\mathcal{P})}$ at the final iteration are obtained as the outputs.

2.4 Active Learning

Active Learning (AL) is a machine learning paradigm that aims to reduce the cost of labeling by iteratively selecting the most informative instances from a large pool of unlabeled data for annotation (Settles, 2009). This approach is based on two assumptions: that different data contribute differently to model performance and that labeled instances can be collected iteratively during training. These assumptions are reasonable in many real-world applications, and empirical results have shown that AL can significantly reduce labeling costs (Zhan, Liu, et al., 2021).

In this section, we present a comprehensive review of the literature on Active Learning

(AL). We begin in Section 2.4.1 with a general overview of AL, providing insights into its different perspectives. Additionally, in Section 2.4.2, we introduce Awesome Active Learning (AAL), an AL knowledge library that we maintain. AAL encompasses the latest research on AL, encompassing both technical advancements and practical applications. Moving forward, in Section 2.4.3, we delve into the most fundamental problem scenarios within AL. We then proceed to explore AL query strategies in detail in Section 2.4.4. Furthermore, in Section 2.4.5, we discuss practical considerations that extend beyond the strategies themselves, yet are vital to AL. Finally, in Section 2.4.6, we provide a summary of the applications of AL in various research fields and industries.

2.4.1 General View of Active Learning

In Section 2.3.1, Definition 3 provides the definition of active learning. However, this definition is overly specific and does not offer a broader understanding of AL. In the literature, AL could be explained from various perspectives, which can be summarized as follows:

1. Problem-oriented perspective: AL is ***an approach aimed at reducing annotation costs***.

2. Human-computer interaction oriented perspective: AL involves ***an interactive labeling manner*** between algorithms and oracles.

3. Technique-oriented perspective: AL is ***a method to evaluate the informativeness of data***, allowing the collection of the most significant instances for the corresponding tasks.

4. Taxonomy-oriented perspective: AL is ***a machine learning setting*** where human experts can be involved.

All of these perspectives are valid and can be utilized to explain AL. However, the core essence of AL lies in **labeling as few informative instances as possible to achieve the**

desired performance. This is due to the high cost associated with labeling in practice, and the fact that not all instances hold equal importance for model performance.

2.4.2 Awesome Active Learning Knowledge Library: Categorization of Strategies, Considerations and Applications (for RQ 0)

Active learning has been studied for decades, resulting in numerous literature reviews and surveys. However, recent reviews and surveys typically focus on specific aspects of active learning, such as the query strategy (Zhan, Liu, et al., 2021), collaboration with deep models (Zhan, Wang, et al., 2022; Ren et al., 2022), or the online setting (Cacciarelli and Kulahci, 2023). Consequently, they fail to provide a comprehensive overview of active learning as a whole. Furthermore, the active learning literature is expanding rapidly, making it challenging to keep track of the latest research progress using the previous taxonomy from Settles, 2009.

To provide an appropriate categorization for current active learning research field, we have developed an open-sourced active learning knowledge library called "Awesome Active Learning" (AAL)[2]. This knowledge library encompasses the latest advancements in both technical developments and real-world applications of active learning. Instead of adopting the traditional survey format, our knowledge library is structured as a hierarchical tree, providing greater flexibility and ease of extension. The papers in the library originate from top-tier conferences or journals, typically selected based on China Computer Federation (CCF) ranking criteria [3] (preferably CCF-B and above). Additionally, we subscribe the citations from several renowned papers, including the most notable surveys (Zhan, Liu, et al., 2021). The primary contribution of our library lies in its comprehensive coverage. Unlike most surveys that tend to focus on single aspects such as strategies or NLP applications, our library encompasses a broader spectrum, including methodologies, limitations, interdisciplinarity, various applications, etc. Furthermore, throughout years of updating process,

[2]https://github.com/SupeRuier/awesome-active-learning
[3]https://www.ccf.org.cn/Academic_Evaluation/By_category/

the categorization is continuously refined, leading to a more comprehensive representation. This library is updated regularly, and we welcome contributions from the community to keep it up-to-date.

The structure of the AAL knowledge library is illustrated in Figure 2.3. Firstly, it consists of a curated collection of AL papers, which is organized at the beginning of the library. Subsequently, active learning is explored from three perspectives: the solved problem, assumptions, and explanation perspectives of AL. Following this, various related materials such as surveys, reviews, tutorials, and benchmarks are presented. To delve into the technical aspects, AAL provides an overview of specific problem settings and scenarios, along with the theoretical studies. In order to apply AL in practical scenarios, the basic assumptions in the simplest AL setting should be relaxed. The corresponding assumptions and solutions are summarized to address these practical considerations. Besides, as a technique for reducing labeling costs, AL finds extensive application in diverse fields such as artificial intelligence, science, and industry. The studies of AL-assisted AI, scientific and industrial applications are organized and presented. Moreover, AAL also lists a compilation of open-source AL libraries and software that are available online. Finally, we provide a list of research groups and scholars actively engaged in AL research.

2.4.3 Basic Scenarios

There are three basic scenarios of AL (Settles, 2009), including pool-based AL, stream-based AL, and query synthesis AL. This division is based on the different origins of the to-be-annotated unlabeled instances.

1. **Pool-based AL**: The unlabeled instances are given as a data pool, and AL algorithms are employed to select informative instances from the pool for querying the oracle. The sequential selection process has been defined in Definition 5.

2. **Stream-based AL**: The unlabeled instances are presented in a sequential order as a stream. assess the informativeness of each incoming instance and make decisions on

Figure 2.3: The structure of the **Awesome Active Learning** knowledge library.

whether to query the oracle for its annotation.

3. **Query Synthesis AL**: This setting requires no unlabeled instances. Instead of requesting annotations for existing instances, AL algorithms generate new instances to be queried.

2.4.4 Query Strategy (Pool-Based Only)

This thesis specifically focuses on pool-based active learning, as it is the most common scenario in practice. Recent surveys by Zhan, Wang, et al., 2022 and Zhan, Liu, et al., 2021 also solely examine pool-based AL. Pool-based AL aims to select a subset of informative labeled instances from a pre-collected data pool in order to maximize the performance of the task model, as described in Definition 4. While some methods can be extended to other scenarios, this section only reviews pool-based AL. The taxonomy used in this section is similar to our AAL knowledge library[4].

Figure 2.4 illustrates the main taxonomy of existing active learning strategies and their associated prominent works. Since we have limited the problem setting to the pool-based scenario, this taxonomy primarily focuses on the perspective of query strategies, which determine the criteria for selecting samples for labeling. The query strategies are classified into four categories: informativeness-based, representativeness-impart, learning to score, and others. In the following subsections, we will provide a detailed introduction to each of these categories.

2.4.4.1 Informativeness-Based

Informativeness-based methods define how much the current inference system is uncertain about the output of the corresponding instance.

Uncertainty Sampling (Lewis and Catlett, 1994) uses the probability output to evaluate the uncertainty of the example. Specifically, the most likely possibility (Least Confident,

[4]https://github.com/SupeRuier/awesome-active-learning/blob/master/contents/pb_classification.md

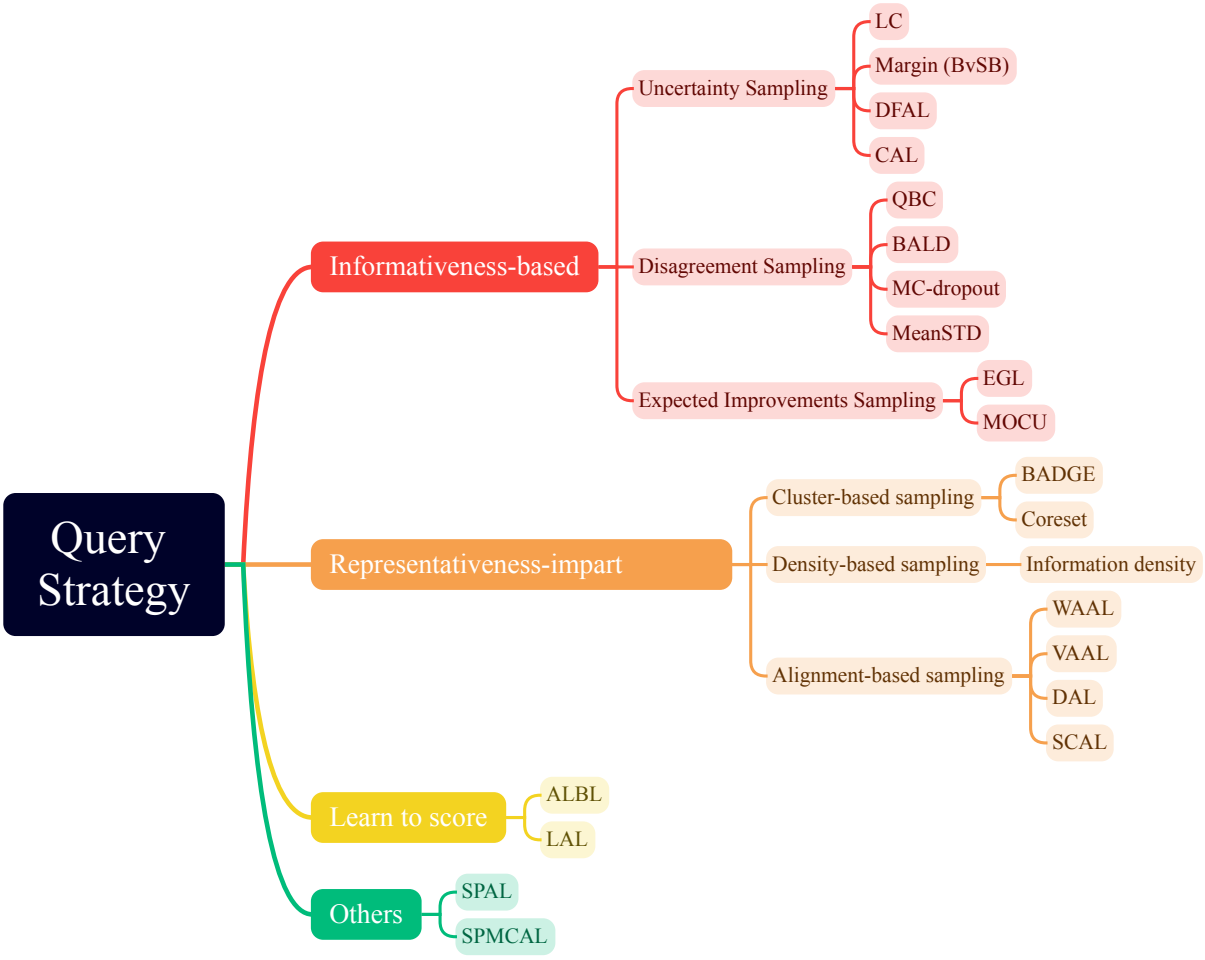


Figure 2.4: The taxonomy of pool-based AL and their associated prominent works.

or LC), the margin between the top-2 classes (Margin or BvSB (Joshi et al., 2009)), and the output entropy (Entropy) can be used to evaluate instances. Take BvSB as an example, the margin is defined as the difference between the probability of the most likely class and the second most likely class. The selection criterion could be defined as $x_{BvSB} = \arg\min_x p_\theta(\hat{y_1} \mid x) - p_\theta(\hat{y_2} \mid x)$. DeepFool active learning (DFAL) (Ducoffe and Precioso, 2018a) proposes to use the distances between the examples and their adversarial examples as the evaluations since measuring the exact distance to the decision boundaries is intractable. Contrastive active leaning (CAL) (Margatina, Vernikos, et al., 2021) considers the inconsistency of predictions with the neighbors as the selection criteria.

Disagreement Sampling uses the disagreement of the outputs from different models as the informativeness evaluation when a bag of models is available in the inference system. Query-by-committee (QBC) (Seung et al., 1992) utilizes the disagreement of multiple classifiers. Bayesian active learning by disagreement (BALD) (Houlsby et al., 2011) is based on a probabilistic Gaussian process classifier. It seeks instances in which the model parameters under the posterior disagree about the outcome the most. Monte-Carlo (MC) dropout (Gal et al., 2017) extends BALD to a Bayesian CNN model, and the prediction uncertainty is induced by marginalizing over the approximate posterior using Monte Carlo integration (models sampled with dropout). Mean standard deviation (MeanSTD) (Kampffmeyer et al., 2016) computes the standard deviation over the softmax outputs of Monte Carlo samples as an uncertainty evaluation.

Expected Improvement Sampling considers the instance that most improves the model's performance as the informative instance. This term is usually heuristically calculated by how much the instance can influence the model, for example, the expected gradient length (EGL) (Settles and Craven, 2008; Zhang, Lease, et al., 2017). The instances with the largest expected gradient length are selected. The selection criterion could be defined as $x^*_{EGL} = \arg\max_x -\sum_i p_\theta(y_i \mid x)\|\nabla l_x(\theta)\|$. Several works maximize the expected loss reduction (Roy and McCallum, 2001) or the expected variance reduction (Schein and Ungar, 2007) in a one-step-look-ahead manner. Some methods aim to reduce the model uncertainty related to the

classification error, such as mean objective cost of uncertainty (MOCU) (Zhao, Dougherty, et al., 2021).

2.4.4.2 Representativeness-Impart

The informativeness-based strategies usually focus on the decision boundary and neglect the data distribution. In this situation, many works take the representativeness of the data into account. Representativeness measures how much the labeled instances are aligned with the unlabeled instances in distribution. We note that representativeness is commonly used with informativeness for sampling.

Cluster-based sampling evaluates instances in a pairwise-matrix form, and no scores are provided for the selection. As a two-stage approach, a pre-selection or a data preprocessing procedure needs to be carried out in advance. Then, the selection can be conducted on the pre-selected informative instances (Xu, Yu, et al., 2003; Citovsky et al., 2021; Zhdanov, 2019) or the induced gradients such as batch active learning by diverse gradient embeddings (BADGE) (Ash, Zhang, et al., 2020). Specifically, in BADGE, the gradients of the model's output with respect to the input are used to measure potential influence on the model as vectors. Clustering is then applied to the gradients, leading to a diverse set of selections. The clustering can also be applied to the original features for further informativeness selection (Nguyen and Smeulders, 2004). Without considering informativeness, Coreset (Sener and Savarese, 2018) formulates AL as a core-set selection. In the representation space, Coreset strategy greedily selects the instance that has the largest distance to the current core set. The model trained on the selected core set can perform as closely as possible to the model trained on the entire dataset.

Density-based and **Alignment-based sampling** provide scores as informativeness-based methods do. Thus, the acquisition function can easily be calculated as $\alpha_{\text{overall}} = \alpha_{\text{informativeness}} + \lambda \times \alpha_{\text{representativeness}}$, where the λ is a trade-off parameter. Density-based sampling points out that the location with more density should be more likely to be queried, e.g., information density (Settles and Craven, 2008). Alignment-based sampling directly considers the distri-

bution alignment between labeled (selected) data and unlabeled data. Discriminative active learning (DAL) (Gissin and Shalev-Shwartz, 2019), variational adversarial active learning (VAAL) (Sinha et al., 2019), wasserstein adversarial AL (WAAL) (Shui et al., 2020), dual adversarial network (DAAL) (Wang, Li, et al., 2020) and MinimaxAL (Ebrahimi et al., 2020) attempt to utilize the distinguishability to evaluate the representativeness. The instances with a higher probability of being unlabeled data will be selected under the evaluation of the discriminator. Supervised contrastive active learning (SCAL) (Krishnan et al., 2021) selects the least similar instances to the labeled ones with the same class in the embedding space.

2.4.4.3 Learn to Score

It is hard to find a single strategy that dominates all the learning tasks. Several works try to learn an acquisition function from the labeling process instead of manually designing the function. Active learning by learning (ALBL) (Hsu and Lin, 2015) utilizes multiple strategies as a multi-armed bandit learner and estimates the performance of different strategies on the fly. Learning active learning (LAL) (Konyushkova et al., 2017) trains a random forest regressor to predict the expected error reduction for a candidate sample at particular learning states. Learning loss for active learning (LLAL) (Yoo and Kweon, 2019) attaches a small parametric "loss prediction module" to a target network for predicting target losses of unlabeled inputs. Some other works formulate the AL process as a Markov decision process and try to solve it as a reinforcement learning (Fang, Li, et al., 2017) or an imitation learning (Liu, Buntine, et al., 2018) problem. Learn from historical sequences (LHS) (Yao et al., 2020) is to learn an active learning ranker from the previous samples to guide the selection.

2.4.4.4 Others

Except informativeness and representativeness, other innovative heuristics can also be involved. Considering the easiness of instances, SPAL (self-paced active learning) (Tang and Huang, 2019) and SPMCAL (self-paced multi-criteria active learning) (Yin et al., 2021) consider AL as a self-paced learning problem and select from the easy instances to the hard

ones.

2.4.5 Practical Considerations Beyond Strategies

In the conventional setting of active learning, the improvement of the entire system is typically attributed solely to the AL strategy. While the query strategy plays a crucial role in the iterations of active learning, it is not the only component involved. Due to varying requirements and assumptions, the AL process can be formulated in different ways. These practical considerations often present challenges in the AL process, requiring the construction of solutions through the use of strategies or other components.

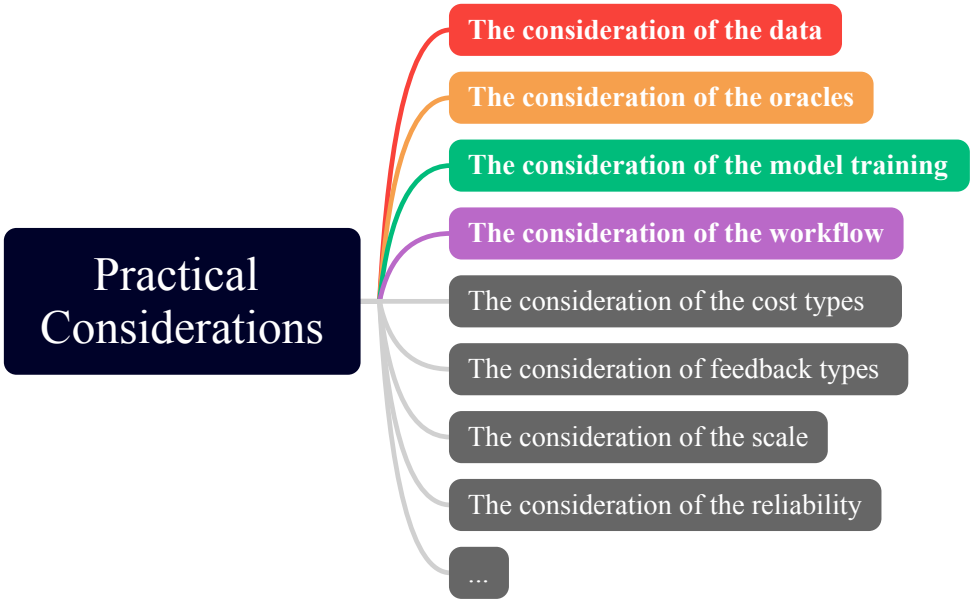


Figure 2.5: The practical considerations for utilizing AL.

The practical considerations encompass a wide range of aspects, as depicted in Figure 2.5. Alongside the strategies, there are three primary components that constitute an AL process: data, oracles, and a model under a specific training paradigm. These terms are integrated within a unified workflow. In this subsection, we will only introduce the most common aspects of the considerations, as indicated by the colored terms in Figure 2.5. For

more detailed information, please refer to the AAL knowledge base[5].

Data augmentation: The data augmentation (Shorten and Khoshgoftaar, 2019b) approaches have been widely used in supervised learning. The augmented data can be used in model training to enlarge the labeled training set, e.g., Bayesian generative active deep learning (BGADL) (Tran et al., 2019). With data augmentation, look-ahead data acquisition (LADA) (Kim et al., 2021) further takes the influence of the augmented data into account to construct the acquisition function.

Model training with unlabeled data: Traditionally, models are trained in a fully supervised manner without additional manipulation. However, several studies have explored the combination of active learning (AL) with semi-supervised learning (SemiSL) (Engelen and Hoos, 2020). Leveraging unlabeled data, these approaches train the model accordingly. Inspired by the wrapper methods in SemiSL, some studies incorporate pseudo-labels in model training (Siméoni et al., 2020), such as cost-effective active learning (CEAL) (Wang, Zhang, et al., 2017). Others utilize data perturbation (Gao, Zhang, et al., 2020) to enhance robustness in semi-supervised AL. The distinguishability of labeled and unlabeled data can also be utilized to train classifiers (Caramalau et al., 2021). Incremental training methods (Krishnan et al., 2021) have been explored as an alternative to retraining the entire model at each AL iteration. Additionally, pretraining schemes have shown promise in improving AL performance (Tamkin et al., 2022; Bengar et al., 2021).

Diverse and imperfect oracles: Conventionally, oracles are assumed to be single and perfect, capable of providing ground-truth labels for any given instance. However, in practical scenarios, oracles can exhibit diversity and imperfection (Fang, Yin, et al., 2013; Zhang and Chaudhuri, 2015). Consequently, the AL process needs to consider which oracle to query, taking into account variations in cost (Huang, Chen, et al., 2017) and quality (Gao and Saar-Tsechansky, 2020).

Batch-mode selection in workflow: Examples are queried in batches in the practical training process to ensure effective training. Conventional active learning tends to greedily

[5]https://github.com/SupeRuier/awesome-active-learning/blob/master/contents/practical_considerations.md

select instances with the top evaluation scores from the acquisition function, leading to information overlap. Batch-mode AL (also called diversity-based AL) tries to solve this myopic problem by diversifying the selected batch. This type of method has huge overlaps with representativeness-impart methods, which unintentionally increase the diversity of the selected batch, such as the two-stage strategies introduced in Section 2.4.4.2, e.g., BADGE (Ash, Zhang, et al., 2020). The diversity can also be intentionally ensured by using greedy selection such as K-center-greedy (Sener and Savarese, 2018), BatchBALD (Kirsch et al., 2019), and BAIT (batch active learning via information matrices) (Ash, Goel, et al., 2021). Active-DPP (Biyik et al., 2019) utilizes Determinantal Point Processes to diversify the batch.

2.4.6 Applications of AL

AL assisted AI: Intuitively, active learning can be effectively integrated with various other fields of AI[6], as most AI tasks heavily rely on labeled data. Numerous surveys have explored the applications of AL in different AI domains, including computer vision (Takezoe et al., 2022), natural language processing (Zhang, Strubell, et al., 2022), recommendation systems (Elahi et al., 2016), and robotics (Taylor et al., 2022).

AL assisted Science: AL naturally finds its place in scientific research due to the scarcity and expense of labeled data. Annotating data usually requires the expertise of domain specialists, experiments, or resource-intensive simulations. AL can be effectively employed in numerous scientific fields[7] such as biology (Borkowski et al., 2020), physics (Teixeira Parente et al., 2023), geology (Chang et al., 2022), etc. By incorporating AL, scientific research becomes more efficient and cost-effective.

AL assisted Industrial Tasks: AL holds potential for various industrial applications as well[8], including autonomous driving (Peng et al., 2021), drug discovery (Ding et al., 2021),

[6]https://github.com/SupeRuier/awesome-active-learning/blob/master/contents/AL_combinations.md

[7]https://github.com/SupeRuier/awesome-active-learning/blob/master/contents/AL_applications.md#scientific-applications-alphabetical-order

[8]https://github.com/SupeRuier/awesome-active-learning/blob/master/contents/AL_applications.md#industrial-applications-alphabetical-order

remote sensing (Tuia et al., 2021), software engineering (Samoaa et al., 2023), etc.

2.5 Learning from Multiple Domains

In this thesis, we focus on the problem of multi-domain learning, as we defined in Definition 6. There are many learning paradigms that also include multiple domains, such as domain adaptation (Pan and Yang, 2010), multi-domain learning (Dredze and Crammer, 2008), and domain generalization (Yuan, Ma, et al., 2022; Wang, Lan, et al., 2021). These paradigms sometimes share similar intuition to extract the domain-shared knowledge, but they face different problem settings. Thus, to make the review more comprehensive, beside the exact MDL setting, we also include the relevant works in other paradigms, where the learning process includes multiple domains. Firstly, we introduce the general cross-domain information-sharing schemes. Then, we review a setting where only limited number of annotations are available in each domain.

2.5.1 Cross-Domain Information-Sharing Schemes

This subsection reviews the cross-domain information sharing schemes in the relevant research fields, such as domain adaptation (Pan and Yang, 2010), multi-domain learning (Dredze and Crammer, 2008), and domain generalization (Wang, Lan, et al., 2021). This subsection is organized in a technique-oriented order. The information-sharing usually assumes that the knowledge from one domain can also assist the learning in another domain. The relevant approaches are classified by the carrier of the information. The primary taxonomy and the representative works of the current cross-domain information-sharing schemes are shown in Figure 2.6.

2.5.1.1 Sharing Instances Across Domains

Directly utilizing the instances from one domain to train a model for another domain is referred to as instance transfer in domain adaptation (Pan and Yang, 2010). The essence is that

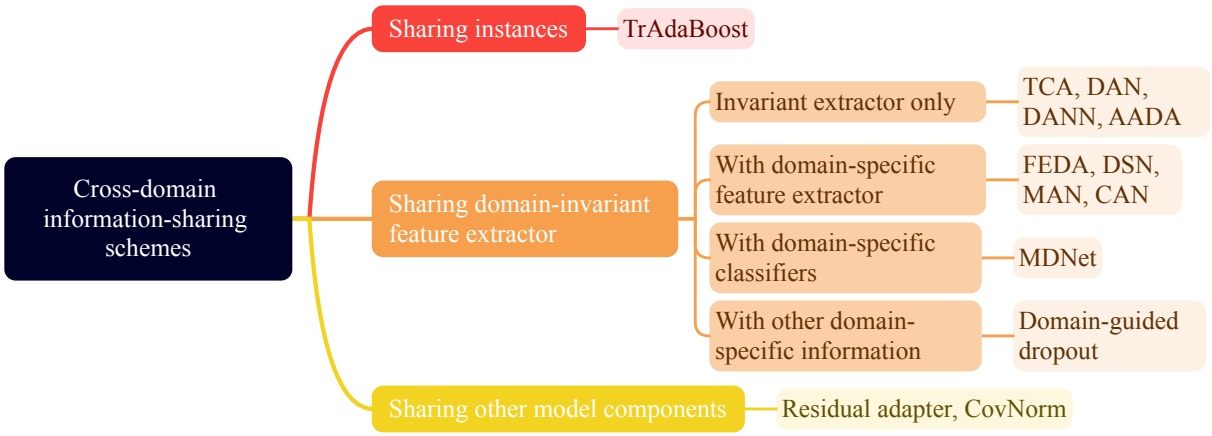


Figure 2.6: The taxonomy for cross-domain information-sharing schemes and the representative works.

some instances can be reused for another domain as they do not contain misleading information. The data from the source domain can be directly used to train the target classifier as auxiliary examples (Wu and Dietterich, 2004). Further, Jiang and Zhai, 2007 removed the misleading instances to train the target model. Instead, transfer adaBoost (TrAdaBoost) (Dai, Yang, et al., 2007) sets weights to source data to balance their contribution to the target domain.

2.5.1.2 Sharing Domain-Invariant Feature Extractor

Distributions from different domains can be unified to eliminate the negative effects of the domain-specific features. A domain-invariant feature extractor can be learned for the subsequent inferences.

Sharing domain-invariant extractors only: Plenty of works match the marginal distribution by minimizing a discrepancy loss, such as the Maximum Mean Discrepancy (MMD). Transfer component analysis (TCA) (Pan, Tsang, et al., 2011) is a classic method that applies this idea to conventional kernel-based models. Deep adaptation networks (DAN) (Long, Cao, et al., 2015) and joint adaptation networks (JAN) (Long, Zhu, et al., 2017) adopt this idea to neural network-based domain adaptation models. Several works utilize adversarial

training to match the marginal distributions. The representation extraction can be effective if a discriminator cannot tell which domain the instances come from. Domain-adversarial neural networks (DANN) (Ganin et al., 2016), adversarial discriminative domain adaptation (ADDA) (Tzeng et al., 2017), and conditional adversarial domain adaptation (CADA) (Long, Cao, et al., 2018) add discriminators into their models for domain adaptation. Feng, Xu, et al., 2019 also adapted this idea to learn representations in MDL.

Some other works proposed to match conditional distributions. The instances in the same class from different domains are expected to be projected closer. Deep supervised domain adaptation (SDA) (Motiian et al., 2017), multi-adversarial domain adaptation (MADA) (Pei et al., 2018), and transferrable prototypical networks (TPN) (Pan, Yao, et al., 2019) match the distributions either by the existing true labels or the created pseudo-labels on the target domain for domain adaptation. Saito et al., 2018 tried to align the distributions by matching the decision boundaries.

Sharing domain-invariant extractors with domain-specific extractors: Solely learning the domain-invariant representations wastes the unique information of each domain from the data. Thus, domain-specific information can further be included to guide the inference. The domain-invariant and domain-specific representations can be concatenated together to make predictions. Frustratingly easy domain adaptation (FEDA) (III, 2007) first applies this idea to MDL with linear models. The widely used neural network architecture for addressing this challenge is the shared-private structure, originally utilized in domain adaptation problems, such as domain separation networks (DSN) (Bousmalis et al., 2016). Adversarial shared-private model (ASP-MTL) (Liu, Qiu, et al., 2017) is the pioneering approach to employ the shared-private structure in the context of MDL, resulting in significant improvements over single-domain models. ASP-MTL adopts adversarial learning to encourage domain-invariance in the shared feature extractor and domain-specificity in the private feature extractors. Several subsequent works have further enhanced the performance based on this share-private architecture, such as multinomial adversarial networks (MAN) (Chen and Cardie, 2018) and conditional adversarial networks (CAN) (Wu, Inkpen, et al., 2021). MAN

includes a universal classifier among different domains. CAN further includes the conditional distribution matching in training the shared extractor.

Sharing domain-invariant extractors with other domain-specific information types: Besides private feature extractors, domain-specific information can also be handled in other forms. Multi-domain convolutional neural networks (MDNet) (Nam and Han, 2016) apply different domain-specific classifiers to the shared representations for the multi-domain visual tracking problem. Similarly, Xiao, Li, et al., 2016 proposed a domain-guided dropout for a multi-domain person re-identification problem. Besides, when there is more than one feature extractor (or channel), the representations (Li, Baldwin, et al., 2019) or channels (Xiao, Gu, et al., 2020) can be weighted differently to make predictions on different domains.

2.5.1.3 Sharing Other Model Components

Instead of explicitly sharing the same feature extractor across domains for representation learning, the modularized model components can be shared for transferring the domain information. Some works (Rebuffi et al., 2017; Rebuffi et al., 2018) design networks with domain-specific residual adapters and remain the rest components the same. Li and Vasconcelos, 2019 involved a domain-specific CovNorm layer instead.

2.5.2 Multi-Domain Learning From Limited Labeled Data

The previous section mainly focuses on the information sharing schemes from fully labeled data. However, in many real-world applications, the labeled data are limited. Thus, learning from limited labeled multi-domain data is a more practical and challenging problem. The key to this problem is to effectively utilize the limited labeled data and abundant unlabeled data to improve the performance.

Contrastive learning (CL) (Chen, Kornblith, et al., 2020) is one of the most prevalent paradigm of utilizing unlabeled data. The general learning paradigm of CL always performs contrasting among different views augmented from the original data (Chen, Kornblith, et al.,

2020; Wu, Fan, Chen, et al., 2022; Khosla et al., 2020). Due to the capability of extracting supervision signal from unlabeled data, CL is explored in a wide range of fields and achieves promising performance (Chen, Kornblith, et al., 2020; Wu, Fan, Chen, et al., 2022; Oord et al., 2018), including multi-domain learning.

Contrastive learning has been applied in domain adaptation, where the learning process is guided by a fully labeled source domain to improve performance on the unlabeled or sparsely labeled target domain. Self-supervised tasks can be directly used to learn a joint shared feature extractor (Sun et al., 2019). Besides, class prototypes (Tanwisuth et al., 2021; Singh, 2021; Li, Liu, et al., 2021) can be generated to align the categorical distributions between the source and target domains. Meanwhile, instance contrastive alignment (Singh, 2021; Yan, Wu, et al., 2022) can also be used to build a feature extractor without using labels on the target domain to learn the target domain representation.

2.6 Annotating from Multiple Domains

In this section, we focus on the setting where the annotators can be collected during the learning process on multiple domains. We first introduce the existing works on cross-domain instance selection, mostly in active domain adaptation setting. Then, we discuss the works on multi-domain instance selection, which is the multi-domain active learning problem for this thesis.

2.6.1 Cross-Domain Instance Selecting Schemes

The AL strategies for active domain adaptation usually select the instances that can minimize the domain discrepancy to learn the domain-invariant representations better. Chattopadhyay et al., 2013 tried to select instances from the target domain by reducing MMD between the labeled and the unlabeled data. Huang and Chen, 2016 utilized this idea in another problem setting, where the selection was from the source domain. Deng et al., 2018 train a multi-kernel SVM classifier with the source and target data, then they used the margin criteria

to select instances. Su et al., 2020 proposed active adversarial domain adaptation (AADA), where a discriminator is trained to weight the uncertainty scores of the target unlabeled examples.

2.6.2 Multi-Domain Active Learning

Selecting instances from different domains for AL requires proper cross-domain instance evaluation criteria. The instances most beneficial to all the domains will be selected, this is referred to as multi-domain active learning. The works in this field are usually strongly related to the models introduced in Section 2.5.1, as the AL strategies under this category are usually model-dependent.

Only a limited number of studies directly relate to MDAL, where the domain-shared representations are used with the domain-specific representations. These works still employ conventional single-domain AL strategies on models trained on multiple domains. For instance, Li, Jin, et al., 2012 first applied active learning with multiple support vector machines on concatenated features from each domain in the context of multi-domain sentiment classification. The instances that can mostly reduce the version space of the SVM models are selected. Zhang, Jin, et al., 2016 selected the user-item pairs leading to the lowest global generalization error of the model in the application of multi-domain recommendation. The application of these studies is inherently limited as they are only tailored for ad hoc tasks on specific types of models. He, Liu, He, et al., 2023 conducted a comprehensive comparative study of conventional active learning strategies on multiple neural-network-based MDL models, demonstrating improvements over random selection. Some works have also applied active learning to multiple domains without considering information-sharing. They either construct independent classifiers for each domain (Vercruyssen et al., 2022) or utilize a single model for all domains (Snijders et al., 2023).

In these existing works, cross-domain information is primarily considered during the model training process, while the selection process evaluates the informativeness of items

solely within specific domains. In other words, current AL researches do not account for the potential impact of a sample on other domains.

2.7 Chapter Summary

This chapter provides the background knowledge of this thesis: (1) The location of our main research problem in the whole machine learning field. (2) The definitions of the main concepts in this thesis. (3) Active learning review and the corresponding works and applications. (4) Approaches for learning from multiple domains. (5) Approaches for annotating from multiple domains.

Chapter 3

Multi-Domain Active Learning: a Comparative Study

In chapter 2, we have introduced the background of active learning (AL), multi-domain learning (MDL) and multi-domain active learning (MDAL). As mentioned earlier, there have been only a few studies conducted on MDAL. Furthermore, to our knowledge, no previous research has explored MDAL with neural networks, which limits the full potential of MDAL. To address this gap, the most intuitive solution is to integrate conventional active learning methods with MDL and neural networks to devise a unified MDAL framework. It is worth noting that the integrated active learning methods should be model-agnostic and adaptable to the structure of MDL and neural networks. Following this approach, there could be a wide range of potential MDAL algorithms, each of which is created by combining different MDL models with various AL strategies. Therefore, a natural question arises: **how do traditional active learning methods perform in multi-domain learning?**

In this chapter, we construct a pipeline for MDAL and present a comprehensive comparative study of thirty different algorithms. These algorithms are created by combining six representative MDL models with five commonly used AL strategies. We evaluate the performance of these algorithms on six datasets that involve textual and visual classification tasks.

The results indicate that in most cases, AL significantly enhances the performance of MDL. Interestingly, the simple best vs. second best (BvSB) uncertainty strategy demonstrates competitive performance compared to state-of-the-art AL strategies. Moreover, when combined with the multinomial adversarial networks (MAN) model, the BvSB strategy consistently achieves top or above-average performance across all datasets. Our qualitative analysis of the well-performing strategies and models sheds light on their superior performance in the comparison. Based on the results, we recommend using BvSB in conjunction with the MAN model for MDAL applications, given their strong performance in the experiments.

The subsequent sections of this chapter are organized as follows: In Section 3.1, we provide additional details about the background and motivation behind this work. Section 3.2 outlines the MDAL pipeline, offering an easy-to-implement solution for MDAL. Moving forward, Section 3.3 describes the experimental settings, followed by the presentation of the comparative study results in Section 3.4. In Section 3.5, we conduct an in-depth analysis of the results and discuss the reasons for the superior performance of certain models and strategies. Finally, Section 3.6 contains a thorough discussion of the results, concluding this chapter.

This chapter has been previously published in the Work 1 (He, Liu, He, et al., 2023). It should be emphasized that this version includes additional experimental runs and enhanced statistical analysis, offering a more thorough evaluation of the proposed method.

3.1 Background and Motivation

Building classifiers on the data collected from different domains is common in real-world applications (Dredze and Crammer, 2008). Here, domains usually refer to different datasets under different distributions. For example, in sentiment analysis of product reviews, distinct product categories are considered as different domains. The conventional approach is to build one single model on all the domains jointly or build models on each domain independently. However, the joint training eliminates the unique information of each domain, and the in-

dependent training neglects the correlation among domains. Thus, both approaches usually bring suboptimal performance. When the number of domains is large, such limitations can be more serious. Under this circumstance, multi-domain learning (MDL) (Dredze and Crammer, 2008) has been proposed to capture both the domain-invariant and the domain-specific information to overcome the aforementioned limitations.

In real life, high labeling effort is generally required in MDL, as data needs to be labeled by human experts for every domain. Active learning (AL) (Settles, 2009) is a general approach to reducing the labeling effort by interactively selecting and labeling the most informative instances in conventional single domain learning. Hence, a natural question arises: can AL be used to reduce the labeling effort in MDL? Despite the practical importance of this problem, only a few studies have been conducted on this issue. The application of these studies is inherently limited as they are only tailored for ad hoc tasks on specific types of models. Specifically, the strategy in (Li, Jin, et al., 2012) needs to evaluate the size of the model's version space (Tong, 2001), which is intractable for models other than SVMs. Besides, the strategy in another work (Zhang, Jin, et al., 2016) is specifically designed for Rating-Matrix Generative Model, which can only be used for the multi-domain recommendation problem. Thus, it is unclear how AL would perform in MDL on more general tasks with more advanced neural networks.

To fill this gap, we study the problem of utilizing AL to reduce the high labeling effort in MDL, which is termed multi-domain active learning (MDAL) (Li, Jin, et al., 2012). On one hand, MDAL keeps the knowledge-sharing in MDL which improves the performance. On the other hand, MDAL tries to maintain the informative labeled set as small as possible to reduce the labeling cost. We have provided an exhaustive literature review on the relevant fields that can serve as the foundations of MDAL in Chapter 2, including AL in Section 2.4, cross-domain information sharing schemes in Section 2.5, and cross-domain instance evaluation approaches in Section 2.6. By drawing insights from this extensive review, we construct an easy-to-implement pipeline of MDAL, which harnesses the power of modern neural network-based MDL models and conventional AL strategies. Subsequently, a compre-

hensive comparative study of thirty different MDAL algorithms is presented. The algorithms are established by combining six representative MDL models and five commonly used AL strategies. We evaluate the algorithms on six datasets involving textual and visual classification tasks. Besides, the well-performed models and strategies are qualitatively analyzed for their superiority. The implementation[1] is available online so that people can easily adopt the pipeline to their MDAL tasks.

The main contributions of this chapter can be summarized as follows:

- We have developed an easily implementable and adaptable pipeline for MDAL, serving as an effective off-the-shelf solution. This pipeline can be applied to various alternative models and strategies for convenient comparisons.
- This study presents the first comparative analysis of MDAL techniques. The results demonstrate that the naive best vs. second best (BvSB) uncertainty strategy performs competitively with state-of-the-art AL strategies. Furthermore, when combined with the multinomial adversarial networks (MAN) model, the BvSB strategy consistently achieves top or above-average performance across all datasets. As a result, we highly recommend utilizing BvSB with the MAN model in the application of MDAL.
- In-depth investigations into the MAN model's performance superiority reveal that the shared-private structure is a key contributing factor. Additionally, we identify that the effectiveness of the uncertainty strategy is linked to the high intra-batch diversity. These findings shed light on the underlying mechanisms of MDAL and provide valuable insights for future research directions.

3.2 Method: The Pipeline for MDAL

We propose a pipeline for MDAL, as shown in Figure 3.1. The procedure is similar to the conventional AL, where the model is trained on the labeled set and the unlabeled set, and

[1] https://github.com/SupeRuier/mdal-pipeline

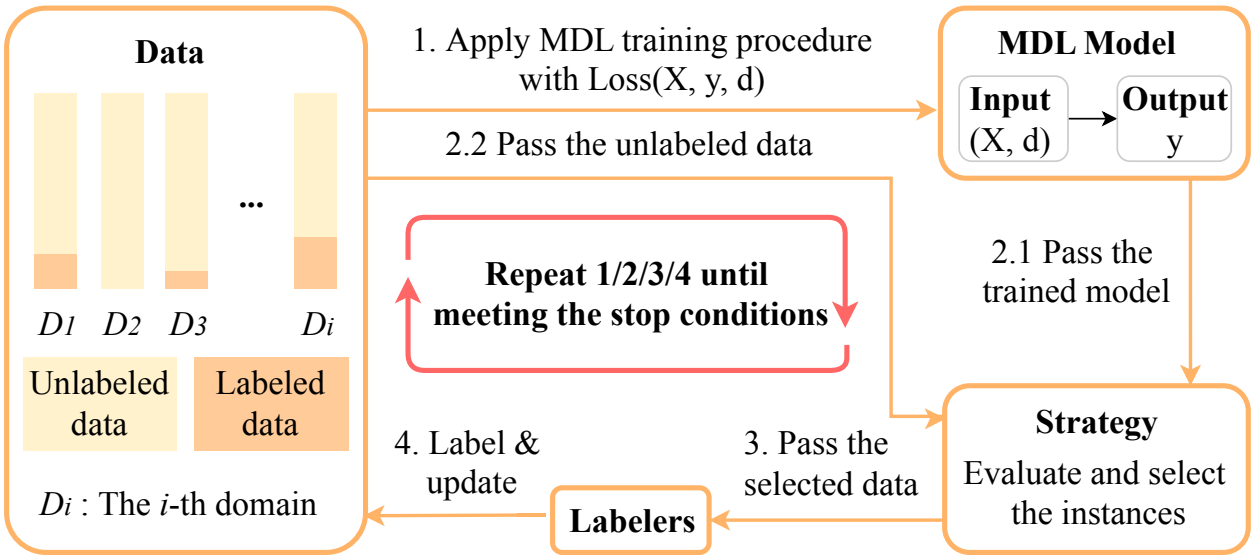


Figure 3.1: The proposed MDAL pipeline combines the MDL models and the conventional AL strategies. Given the current labeled set, the MDL model is trained on the labeled set and the unlabeled set. Then, the AL strategy selects and annotates instances from the unlabeled set by evaluating the instances with the trained MDL model. The evaluation is based on the models' output or the corresponding representations, depending on the specific AL strategy. Instances from different domains are selected simultaneously based on the evaluations. The newly annotated instances are added to the labeled set, and the process iterates until the budget is depleted.

the AL strategy selects and annotates instances from the unlabeled set by evaluating the instances with the trained model. After that the newly annotated instances are added to the labeled set, and the process iterates until the budget is depleted. Compared to the conventional single-domain AL, the differences are in data formulations, model structures & training, and AL strategy. The data are collected from multiple domains, with different number of annotations. Besides, MDL models in this pipeline are designed to capture the difference among the domains. The AL strategy is designed to select instances from different domains simultaneously, instead of only a single domain.

The **model structures & training** and **AL strategy** are two most important components in the pipeline, which could be properly designed and switched. Combining the models for MDL and the conventional AL strategies in an iterative manner can be utilized as an off-the-shelf solution for MDAL. The combination makes this pipeline easy to be implemented and adaptable to many alternative models and strategies.

It should be noted that conventional AL strategies can be used in our MDAL pipeline to select the instances from different domains. However, due to the division of domains, the AL strategies need to be modified according to the types of applied strategies. Specifically, (1) the score-based strategies calculate a score for each instance as the evaluation. The scores from different domains are gathered together and ranked, and the instances with the top-K scores are selected. (2) The two-stage strategies select instances without calculating their scores. Instead, they obtain the gradients or the embeddings of the instances as evaluations in the first stage. Then, the evaluations from different domains are gathered together as a unified pool. The strategies select instances from the pool in the second stage, as they usually do in conventional single domain AL.

3.3 Design of Comparative Experiments

This section describes the research questions and setups of the comparison experiments, given the MDAL pipeline. First, the research questions for the MDAL pipeline are intro-

duced in Section 3.3.1. The selected datasets, models, and AL strategies are introduced in Section 3.3.2, 3.3.3 and 3.3.4, respectively. Section 3.3.5 describes the details of implementations for model training and AL procedure. The performance metrics are introduced in Section 3.3.6.

3.3.1 Research Questions

With this pipeline, several key questions arise:

1. Firstly, we need to conduct a preliminary comparison of multi-domain approaches (information-sharing schemes) to check which approach generally performs better. (Section 3.4.1)

2. Additionally, we seek to assess whether AL can yield improvements over random selection. At the same time, we are interested in identifying any specific strategy that significantly outperforms the others. To achieve this, a comprehensive comparison among all model-strategy pairs is necessary. (Section 3.4.2)

3. Moreover, we are interested in investigating whether the models and strategies that exhibit strong overall performance maintain consistent effectiveness across each domain. This calls for a domain-wise performance analysis of the well-performing model-strategy pairs. (Section 3.4.3)

To address these questions adequately, a series of thorough comparisons must be conducted. This involves comparing different models and strategies on various datasets within the pipeline. By doing so, we can gain valuable insights into the most suitable models, the effectiveness of AL strategies, and the domain-specific performance of the selected model-strategy combinations.

3.3.2 Datasets

Six datasets are selected from the literature on MDL and domain adaptation problems. These datasets at least contain two domains. All the tasks are classification tasks, and the categories are the same across domains. The number of instances and the train-validation-test partitions are also presented.

- **Amazon** contains four domains: books, dvd, electronics, and kitchen. The raw sentences are processed by marginalized denoising autoencoders (mSDA) (Chen, Xu, et al., 2012), instances are encoded as vectors with length 30000. For each domain, there are 2000 training samples, 1000 validation samples. There are 3465/2586/4681/4945 test samples on domain 'books', 'dvd', 'electronics', and 'kitchen', respectively.

- **Office-31** (Gong et al., 2012) contains thirty-one categories from three domains: Amazon, Webcam, and DSLR. Briefly, the DeCaf representations (Donahue et al., 2014) are used, the images are encoded as vectors with length 4096. There are 2817/498/795 samples on domain 'amazon', 'dslr', and 'webcam', respectively. The ratio 6:2:2 is used to split the training/validation/test sets.

- **Office-Home** (Venkateswara et al., 2017) contains four distinct domains: Art, Clip-Art, Product, and Real-world. The raw image are preprocessed by Resnet-50 He, Zhang, et al., 2016 and encoded as vectors with length 2048. Each of the four domains has sixty-five categories. There are 2427/4365/4439/4357 samples on domain 'Art', 'Clipart', 'Product', and 'RealWorld', respectively. The ratio 6:2:2 is used to split the training/validation/test sets.

- **ImageCLEF**[2] is a well-balanced dataset containing twelve categories from three domains: caltech, pascal, and imagenet. Each image is encoded as a vector with length 1024. There are 600 samples on each domain, respectively. The ratio 6:2:2 is used to split the training/validation/test set.

[2] https://www.imageclef.org/2014/adaptation

- **Digits** is used in (Ganin et al., 2016), which contains two domains (sub-datasets): MNIST (LeCun, Bottou, et al., 1998) and MNIST-M. MNIST-M is generalized by blending digits from the MNIST dataset over patches randomly extracted from color photos from BSDS500 (Arbelaez et al., 2010). For each domain, there are 50000 training samples, 10000 validation samples and 10000 test samples. Each instance is a digit image with dimension (28,28,3).

- **PACS** (Li, Yang, Song, et al., 2017) is an image dataset that contains seven categories from four domains: art-painting, cartoon, photo, and sketch. The split can be found in the following link[3]. There are 1840/2107/1499/3531 instances on the training set, 208/237/171/398 instances on the validation set, and 2048/2344/1670/3929 instances on the test set for each domain, respectively. Each instance is an RGB image with dimension (227,227,3).

3.3.3 Models

Two baseline models, **SDL-separate** and **SDL-joint**, are included in the study. These models do not employ specifically designed information sharing schemes and use vanilla neural networks. In SDL-separate, multiple networks are independently trained on each domain. This baseline serves the purpose of demonstrating the performance when no information sharing takes place among the domains. On the other hand, SDL-joint neglects the concept of domain and involves training a single neural network on the combined dataset from all domains. This baseline is employed to illustrate the performance when information sharing is considered in an implicit manner.

Most MDL models facilitate information transfer across domains by utilizing domain-invariant extractors, as explained in Section 2.5.1.2. For the purpose of comparison, we have chosen four representative models: **DANN** (Ganin et al., 2016), **MDNet** (Nam and Han, 2016), **MAN** (Chen and Cardie, 2018), and **CAN** (Wu, Inkpen, et al., 2021). These

[3] https://drive.google.com/drive/folders/0B6x7gtvErXgfUU1WcGY5SzdwZVk?resourcekey=0-2fvpQY_QSyJf2uIECzqPuQ

selected models effectively demonstrate how shared and private information is managed, encompassing a wide range of current information-sharing strategies. Moreover, it is worth noting that these models, or the underlying schemes, are not constrained to specific network structures outlined in their original papers. They can be adapted to alternative backbone neural networks. This adaptability makes it convenient to implement these models across various tasks and datasets, enhancing their applicability.

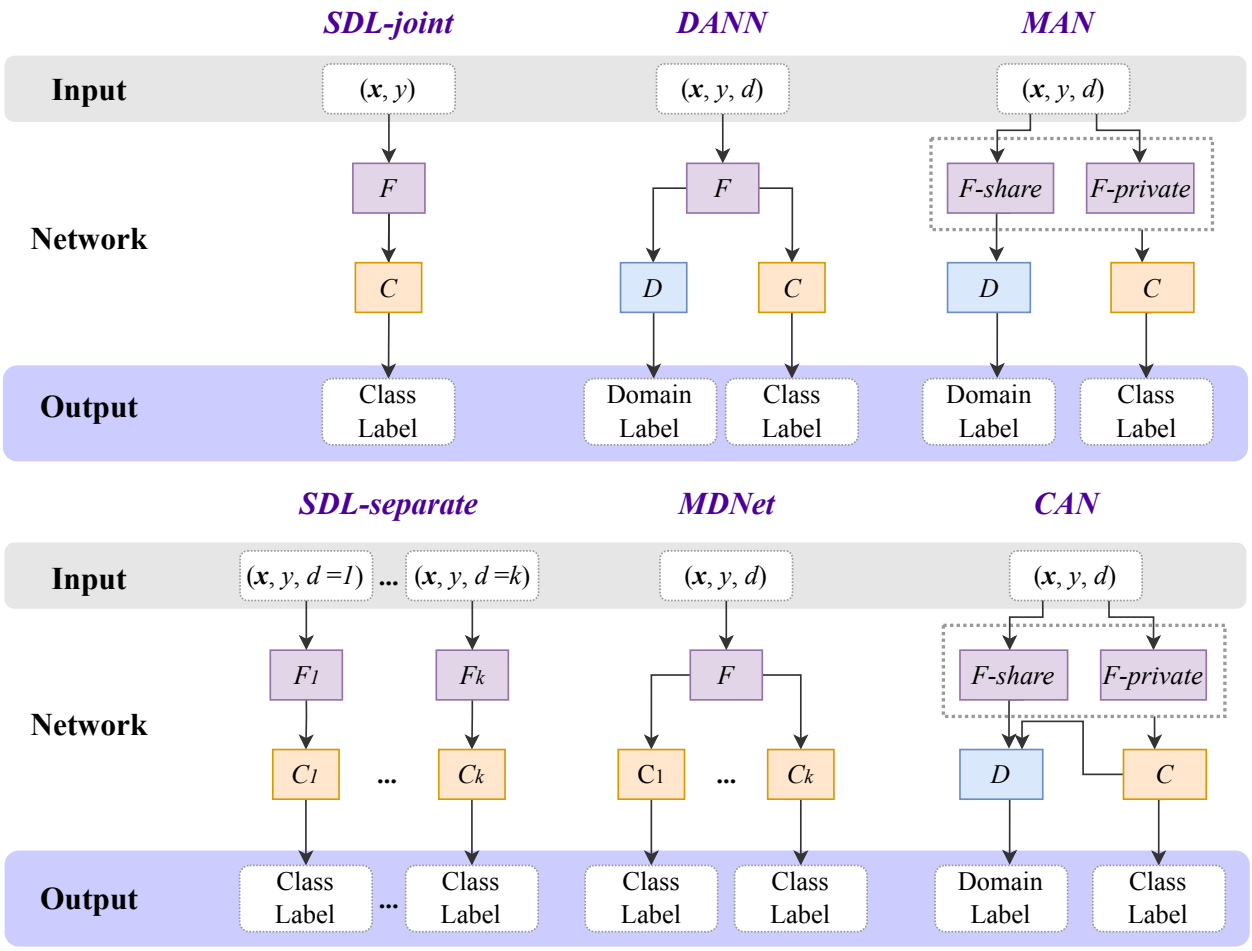


Figure 3.2: The sketches of different models: F represents feature extractors. D represents domain discriminators. C represents classifiers. $\boldsymbol{x}$, y, and d represent the input features, the labels of instances, and the domain label (which domain the item comes from) of the instances, respectively. The feature extractors, discriminators and classifiers are certain neural network structures, which take the input features and output the features, the domain predictions, and the class predictions. In MAN and CAN, the outputs of shared and private feature extractors are concatenated before the class predictions.

Neural networks serve as the backbone model in this study, chosen for their exceptional ability in representation extraction. While the original paper introduced model structures, they were not applied precisely in our case due to variations in input dimensions. Nonetheless, the macrostructures of the models remain consistent with the original paper, where feature extractors (F), classifiers (C), and discriminators (D) are organized following the schematic in Figure 3.2. The microstructures of different models are consistent when applied to the same dataset. Modules such as F, C, or D within different models share the same depth and number of neurons. By employing neural networks and maintaining the core architecture of the original models, we aim to leverage their potent representation extraction capabilities while ensuring the adaptation to our specific input dimensions. The uniformity of microstructures within the models enhances the comparability on the given datasets. Further details of model structure are available in Appendix A.

3.3.4 Strategies

Many of the strategies mentioned in Section 2.6 are unsuitable for our comparison, despite their relevance to various domains. This is because they either rely heavily on non-neural network models or are limited to selecting instances solely from a single target domain for domain adaptation purposes. As a result, we have opted to include five conventional single domain active learning strategies in the MDAL pipeline. These strategies can be readily implemented on neural networks. The characteristics and taxonomy of the compared strategies are shown in Table 3.1.

- **Random**: Randomly select instances from each domain. It serves as a strong baseline.
- **Uncertainty**: There are many uncertainty-based strategies, and Best vs. Second Best (BvSB) (Joshi et al., 2009) usually performs well in multi-class classification tasks. It selects instances with the greatest difference in prediction probability between the most and second most likely classes.

Table 3.1: The characteristics and taxonomy of the selected strategies. Informativeness (Info.), representativeness (Rep.), batch & diversity (Div.) and two-stage selection are considered.

Strategy	Info.	Rep.	Batch & Div.	Two-Stage
Random	-	-	-	-
Uncertainty	✓	-	-	-
EGL	✓	-	-	-
Coreset	-	✓	✓	✓
BADGE	✓	✓	✓	✓

- **EGL** (Settles and Craven, 2008; Zhang, Lease, et al., 2017): Expected Gradient Length is designed for models that can be optimized by gradients. The instances leading to the longest expected gradient length to the last fully connected layer will be selected.
- **Coreset** (Sener and Savarese, 2018): Coreset selects instances using a greedy furthest-first search conditioned on the currently labeled examples. The distance is calculated by using the output of the penultimate layer from the model.
- **BADGE** (Ash, Zhang, et al., 2020): Batch Active learning by Diverse Gradient Embeddings calculates the gradients of the last fully connected layer. A k-means++ initialization is applied to the gradients to ensure the representativeness and diversity of the selected batch.

3.3.5 Details of Implementations

3.3.5.1 Neural Network Training

In the conducted research, both Adam and SGD optimizers were employed to train the models. The training process involved the use of the weight decay term alongside the cross-entropy loss. To strike a balance between the loss of classifiers and discriminators for DANN, MAN, and CAN, a trade-off parameter was carefully determined. Learning rate decay was applied to the optimizers to ensure the stability of the training process. Additionally, the

Table 3.2: The hyperparameters used for the model training.

Datasets	Optimizer	Learning Rate	Learning Rate Decay	Batch Size	Weight decay	Early Stopping	Discriminator Tradeoff
Amazon	Adam	1e-4	-	128	0.05	10	0.1
Office-31	Adam	3e-3	0.333	128	0.001	30	0.1
Office-Home	Adam	1e-4	-	128	0.001	10	0.1
ImageCLEF	Adam	3e-3	0.333	32	0.001	25	0.1
Digits	SGD	1e-2	0.1	128	0.001	15	0.1
PACS	SGD	1e-3	0.1	32	0.001	15	0.1

early stopping technique was implemented during training to mitigate the risk of overfitting. The detail of model structures could be found in the supplementary materials. Deep neural networks are used for Digits and PACS datasets under raw image features, and shallow neural networks with a single hidden layer are used for the other datasets. To ensure a fair comparison, most of the hyperparameters were set uniformly across all models on the same dataset.

Due to the extensive experimentation involved in this work, we try to minimize the burden of hyperparameter selection as much as possible. Specifically, in terms of model architecture, we aligned our choices closely with existing models, ensuring uniformity across network layers, width, and training parameters. For certain hyperparameters without established benchmarks, such as weight decay, we conducted limited grid searches. Our primary aim was not to optimize for a specific evaluation metric, but to ensure a stable overall training process. Observations indicated that variations in hyperparameters had minimal impact on the relative ranking of models and active learning strategies, though they significantly affected absolute performance metrics. Given our focus on comparative experiments, relative performance was deemed more crucial, leading us to avoid extensive hyperparameter optimization. These hyperparameters, instrumental in guiding the model training, are explicitly documented in Table 3.2.

3.3.5.2 AL Setting

Table 3.3: The hyperparameters used for the AL procedure.

Datasets	Total Budget	Initial Labeled Size	AL Batch Size
Amazon	8000	1000	1000
Office-31	2400	400	400
Office-Home	9000	1000	2000
ImageCLEF	1080	180	180
Digits	18000	2000	4000
PACS	8500	500	2000

As a pool-based active learning scenario, the MDAL learning loop iterates until the budget is depleted. A small set of labeled instances (compared to the budget) is randomly selected at the beginning for training an initial model as a warm start. Relatively large AL batch sizes are set to reduce the overall computation burden, because the smaller the AL batch size, the more times of re-training (more iterations) are performed. All the AL process repeats 10 times to obtain an average performance with a standard deviation. The hyperparameters used for the AL procedure are listed in Table 3.3.

3.3.6 Evaluation Metrics

3.3.6.1 Learning Curves

In the MDAL process, the performance of each model-strategy combination is presented as a learning curve. This curve tracks the performance at various labeling budgets, with the domain micro-average accuracy on the test set as the recorded metric. The horizontal axis represents the current budget expense, indicating the number of labeled instances, while the vertical axis illustrates the average performance obtained from multiple runs. Specifically,

the domain micro-averaged accuracy over domains can be calculated as follows:

$$\text{Domain Micro-average Accuracy} = \frac{\sum_{i=1}^{K} \text{Correct Predictions in Domain } i}{\sum_{i=1}^{K} \text{Total Predictions in Domain } i} \tag{3.1}$$

where K is the total number of domains. Given certain number of labels and the corresponding accuracy, the learning curve can be plotted.

It is worth to note that using accuracy as the evaluation metric can be biased when the number of items among domains are very imbalanced. For most of the datasets in this study (also in the thesis), the number of items among domains are relatively balanced, and the item among classes are also balanced. Thus, accuracy is a fair metric to evaluate the performance of the model-strategy pairs. For the relatively imbalanced office-31 and office-home datasets, since the mission is multi-class classification, there also are several famous works utilizing accuracy as the evaluation metric in domain adaptation setting (Ganin et al., 2016; Liu, Long, et al., 2019; Huang, Wen, et al., 2023). Under this circumstance, we use accuracy as the evaluation metric in the whole comparison for simplicity and unity.

3.3.6.2 Area Under the Learning Curve (AULC)

The area under each learning curve can serve as an approximate performance evaluation metric. AULC is particularly useful when comparing numerous combinations simultaneously, since there might be overlaps between the learning curves of different model-strategy pairs. The AULC is calculated by the trapezoidal rule, which is a common method to calculate the area under the curve. Besides, to ensure fair and clear comparisons, the labeling budget is normalized, resulting in a final AULC score is a positive performance value that ranges from 0 to 1. The normalization is performed by dividing the original area by the total adding labels in the AL process. As AULC cannot provide a direct interpretation of the all-stage performance, is it used as a supplementary metric to the learning curves. Besides, the absolute value of AULC is not important, but the relative order of AULC in the comparison is important.

All the AULC results from model-strategy pairs are analyzed using the Mann-Whitney U test (Mann and Whitney, 1947). The Mann-Whitney U test is a non-parametric test, which is used to determine whether there is a statistically significant difference between the medians of two independent samples. The corresponding p-values can be found in appendix A.3. It is worth to note that although two model-strategy pairs might have similar AULC values with small p-values, the learning curves of two methods still can be different, since only the area under the curve is considered in the AULC. Thus, the learning curves need to be checked with the AULC results and the p-values.

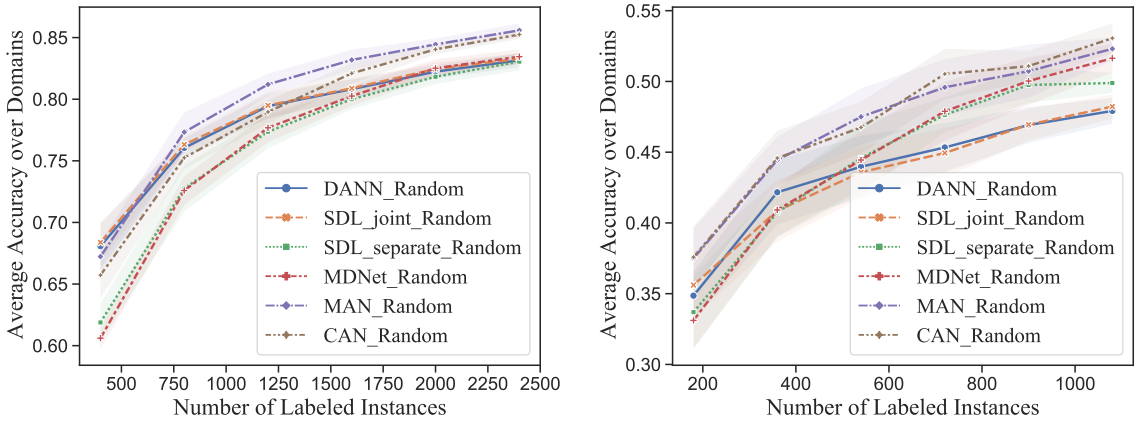
3.4 Results and Analysis

We conduct the following comparisons to answer the three research questions mentioned in Section 3.2. First, the models equipped with different information-sharing schemes are compared in Section 3.4.1. Then, Section 3.4.2 compares the AL strategies on each model according to their total performance over domains. Finally, the per-domain performance is discussed in Section 3.4.3.

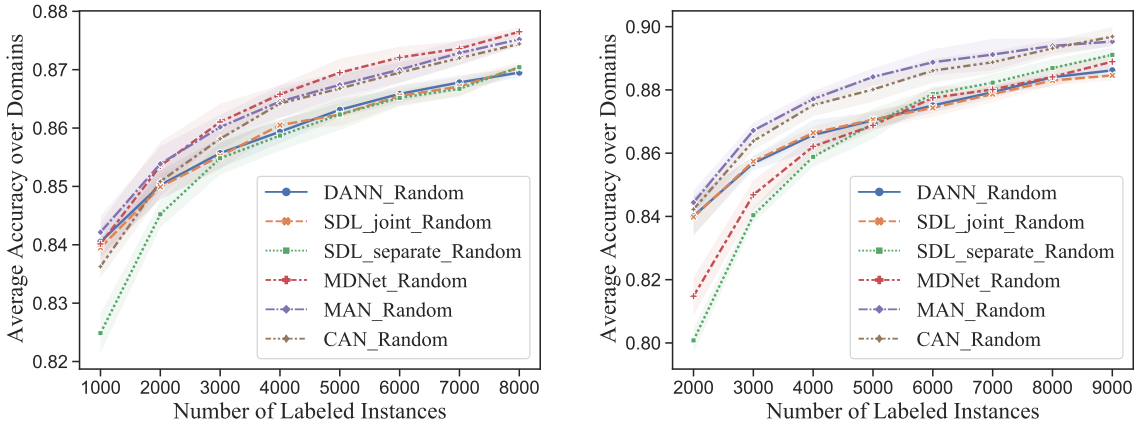
3.4.1 Comparisons over Models

In this section, we conduct a preliminary comparison of multi-domain approaches (information-sharing schemes) to check which approach generally performs better. Since the final performance of MDAL methods is contributed by both the models and the strategies, a top-performed model under random selection is more likely to be a good benchmark for the following comparisons. Thus, in this section, the models with different information-sharing schemes are compared with randomly selected labeled training instances. A model that maintains a good performance in all stages of the whole learning process is desired.

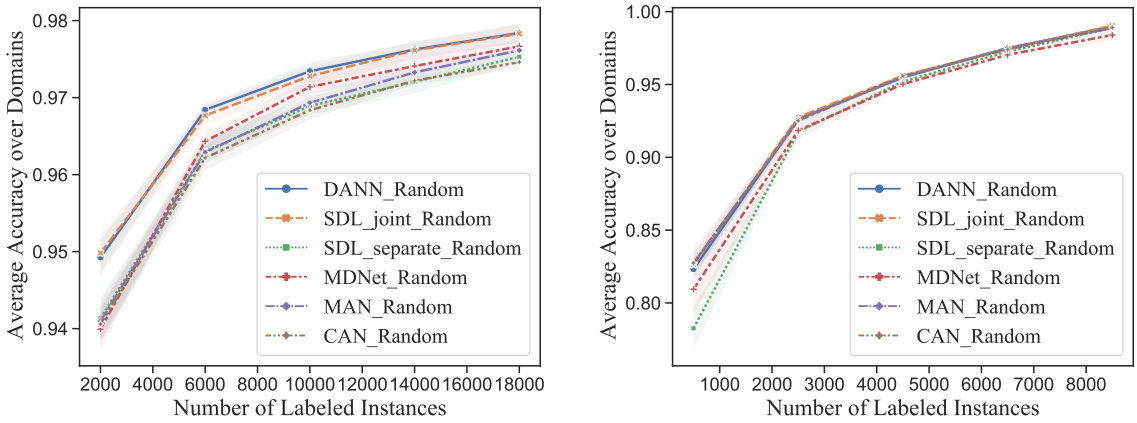
Shallow neural networks with a single hidden layer are used for Office31, ImageCLEF, Amazon, and Office-Home datasets. The features are encoded as vectors in these datasets. The results on these datasets are plotted in Figure 3.3(a)-(d). It is easy to find the similar



(a) Office31 (b) ImageCLEF



(c) Amazon (d) Office-Home



(e) Digits (f) PACS

Figure 3.3: The results of different models on six datasets. Models are compared under different number of randomly selected labeled training instances. The lighter area represents the standard deviation.

trends from the results on these four datasets. The whole inference structures of DANN and SDL-joint are trained on labeled instances from all the domains. Their learning curves have similar trends, and they usually perform well in the beginning but cannot outperform the others in the end. This is because these two models are relatively well-trained with more labeled instances in the beginning compared to other models. SDL-separate and MDNet both have domain-specific classifiers. They usually do not perform well initially, but in the end, they can reach or outperform other models. In the beginning, the domain-specific classifiers can hardly be well-trained with insufficient labeled instances from the corresponding domains, which brings bad initial performances. Moreover, MDNet sometimes outperforms SDL-separate too much, as shown in Figure 3.3(c). It comes from the shared extractor and the simpleness of the binary classification task. MAN and CAN both have share-private inference structures. Their curves also have similar trends, and their performances are usually good during the whole learning process. We believe this superiority comes from the share-private structure, which captures the shared information well in the beginning and maintains the domain-private information in the end. Besides, compared to MAN, CAN sometimes performs clearly worse initially, as shown in Figure 3.3(a) and (c). This is because its conditional distribution matching part cannot be well-trained at the beginning, leading to a negative effect. The corresponding statistical analysis on the AULC can be found in appendix A.3, from Table A.2 to Table A.5. The results of the statistical analysis also support the above analysis.

Deep neural networks are used for Digits and PACS datasets under raw image features. The results are plotted in Figure 3.3 (e) and 3.3 (f). We note that on the PACS dataset, the pretrained ResNet18 (He, Zhang, et al., 2016) is adopted as the feature extractor. It is easy to find that SDL-joint and DANN perform best and SDL-separate performs worst in both datasets. MAN and CAN almost perform the worst on the Digits dataset, but they can obtain a top performance on the PACS dataset. This is because the deep model is less likely well-trained in the beginning without a good initialization (pretrained model) when the labeled instances are very few. The corresponding statistical analysis on the AULC can

be found in appendix A.3, in Table A.6 and Table A.7. The results of the statistical analysis also support the above analysis.

In short, for the first research question, considering the whole learning process, MAN consistently performs well on the shallow networks, and SDL-joint and DANN perform well on the deep networks with the random selected training instances. With pretrained models, MAN can also obtain the top performance with the deep model.

3.4.2 Comparisons over Strategies

All model-strategy combinations are compared on all six datasets in this section. We are curious about whether AL can bring improvements over the random selection and whether there is any strategy that significantly outperforms the others. The performances on six datasets are presented in Table 3.4 in terms of AULC. The raw learning curves can be found in the supplementary materials. From the results, AL brings clear improvements to almost all the datasets. However, it is hard to pick one of the strategies that significantly outperforms the others. The corresponding statistical analysis on the AULC can be found in appendix A.3, from Table A.8 and Table A.13.

Uncertainty, as the most naive AL strategy, consistently obtains good performances in terms of AULC and the learning curves. On most datasets and models, Uncertainty performs competitively with other state-of-the-art strategies. It is easy to find that the p-values between Uncertainty and other top-performed strategies are usually larger than 0.05 under the same model, which means that the differences are not significant. The existence of multiple domains might diversify the selection and improve the performance of Uncertainty (this will be further analyzed in section 3.5). Besides, as a score-based strategy, Uncertainty evaluates the instances solely by the outputs of the models and needs no additional gradients or representations, making it the quickest strategy. Thus, we can claim that Uncertainty is the best strategy for the MDAL pipeline, considering the performance and the evaluation time.

Table 3.4: The area under the learning curve for each model-strategy pair on each dataset. The largest AULC value is in **bold**.

Datasets	Strategy	Models					
		SDL-joint	DANN	SDL-separate	MDNet	MAN	CAN
Office-31	Random	79.00 ± 0.44	78.83 ± 0.42	76.88 ± 0.83	77.01 ± 0.57	80.51 ± 0.48	79.17 ± 0.41
	Uncertainty	79.98 ± 0.44	80.12 ± 0.41	78.80 ± 0.49	79.03 ± 0.37	81.94 ± 0.42	80.50 ± 0.62
	BADGE	79.98 ± 0.46	80.00 ± 0.52	78.65 ± 0.45	78.58 ± 0.34	$\mathbf{81.95 \pm 0.37}$	80.46 ± 0.71
	EGL	79.97 ± 0.60	79.80 ± 0.36	78.17 ± 0.22	78.40 ± 0.48	81.49 ± 0.24	80.01 ± 0.58
	Coreset	80.06 ± 0.44	79.94 ± 0.44	79.25 ± 0.38	77.84 ± 0.50	81.84 ± 0.31	80.39 ± 0.47
Amazon	Random	85.94 ± 0.10	85.96 ± 0.07	85.72 ± 0.13	86.48 ± 0.13	86.39 ± 0.13	86.24 ± 0.12
	Uncertainty	86.38 ± 0.05	86.35 ± 0.05	86.04 ± 0.07	$\mathbf{86.94 \pm 0.09}$	86.88 ± 0.03	86.66 ± 0.07
	BADGE	86.34 ± 0.05	86.34 ± 0.08	85.96 ± 0.09	86.88 ± 0.07	86.84 ± 0.07	86.60 ± 0.07
	EGL	86.37 ± 0.06	86.38 ± 0.06	86.03 ± 0.06	86.93 ± 0.10	86.89 ± 0.06	86.69 ± 0.05
	Coreset	86.26 ± 0.08	86.26 ± 0.05	85.95 ± 0.08	86.83 ± 0.09	86.74 ± 0.05	86.41 ± 0.06
Office-Home	Random	87.04 ± 0.13	87.06 ± 0.13	86.61 ± 0.21	86.73 ± 0.17	88.17 ± 0.19	87.95 ± 0.21
	Uncertainty	87.79 ± 0.10	87.78 ± 0.10	87.79 ± 0.08	87.80 ± 0.07	88.91 ± 0.06	88.76 ± 0.10
	BADGE	87.75 ± 0.06	87.77 ± 0.11	87.74 ± 0.02	87.66 ± 0.11	$\mathbf{88.96 \pm 0.08}$	88.75 ± 0.08
	EGL	87.86 ± 0.13	87.82 ± 0.08	87.70 ± 0.06	87.77 ± 0.09	88.85 ± 0.08	88.77 ± 0.04
	Coreset	87.78 ± 0.12	87.72 ± 0.07	87.82 ± 0.09	87.53 ± 0.14	88.96 ± 0.11	88.65 ± 0.09
ImageCLEF	Random	43.65 ± 0.71	43.96 ± 0.80	44.90 ± 1.02	45.13 ± 1.13	47.43 ± 0.99	47.65 ± 0.81
	Uncertainty	43.44 ± 0.58	43.73 ± 0.71	45.19 ± 0.95	45.52 ± 0.66	47.33 ± 0.86	47.59 ± 1.08
	BADGE	43.91 ± 0.97	44.14 ± 0.78	45.20 ± 0.98	45.40 ± 0.83	47.71 ± 0.58	$\mathbf{48.45 \pm 1.16}$
	EGL	43.30 ± 0.79	43.18 ± 0.55	44.01 ± 1.38	44.50 ± 0.51	46.97 ± 0.74	46.55 ± 1.03
	Coreset	43.52 ± 1.13	43.42 ± 0.79	44.85 ± 0.95	44.36 ± 0.70	47.34 ± 0.80	48.05 ± 1.13
Digits	Random	97.02 ± 0.09	97.05 ± 0.05	96.56 ± 0.07	96.70 ± 0.06	96.60 ± 0.05	96.51 ± 0.07
	Uncertainty	97.95 ± 0.06	97.96 ± 0.05	97.70 ± 0.06	97.57 ± 0.08	97.75 ± 0.04	97.64 ± 0.06
	BADGE	97.98 ± 0.05	$\mathbf{98.00 \pm 0.05}$	97.71 ± 0.04	97.63 ± 0.06	97.75 ± 0.05	97.65 ± 0.06
	EGL	97.98 ± 0.06	97.97 ± 0.05	97.67 ± 0.06	97.61 ± 0.04	97.72 ± 0.05	97.62 ± 0.05
	Coreset	97.76 ± 0.05	97.78 ± 0.05	97.45 ± 0.06	97.42 ± 0.06	97.48 ± 0.07	97.40 ± 0.06
PACS	Random	94.15 ± 0.08	94.06 ± 0.07	93.19 ± 0.17	93.39 ± 0.31	94.10 ± 0.05	94.08 ± 0.21
	Uncertainty	$\mathbf{96.28 \pm 0.10}$	96.27 ± 0.10	95.16 ± 0.29	95.82 ± 0.28	96.07 ± 0.17	96.03 ± 0.36
	BADGE	96.18 ± 0.12	96.08 ± 0.14	95.04 ± 0.28	95.83 ± 0.27	95.86 ± 0.19	95.75 ± 0.31
	EGL	96.24 ± 0.13	96.15 ± 0.11	94.69 ± 0.52	95.82 ± 0.30	95.88 ± 0.10	95.75 ± 0.35
	Coreset	94.41 ± 0.33	93.41 ± 0.31	93.34 ± 0.68	93.24 ± 0.53	94.35 ± 0.35	94.01 ± 0.81

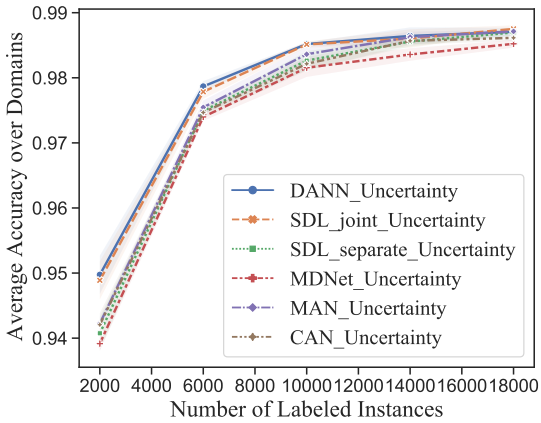
In addition, the final performance of MDAL methods is contributed by both the models and the strategies. Considering the models, Uncertainty with MAN can achieve top performance on most datasets. Intuitively, the combination of the well-performed model and strategy is reasonably more likely to perform better than others. Even though MAN performs poorly with the random selection on the Digits dataset, with Uncertainty, MAN can still obtain a medium performance, as shown in Figure 3.4(a).

In short, no strategy significantly outperforms the others. However, the naive computationally efficient Uncertainty strategy can perform competitively with other state-of-the-art strategies. Especially with the MAN model, Uncertainty can consistently obtain top performance in most cases.

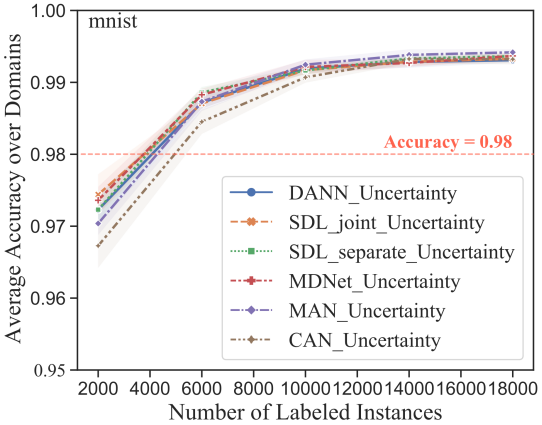
3.4.3 Comparisons over Domains

In this section, we try to find out whether the model-strategy combinations that obtain the excellent overall performance in the previous section can consistently perform well in each domain. Although the overall performance is primarily concerned in MDAL, a model-strategy combination that performs significantly worse on several domains is not acceptable. Specifically, the performance of the Uncertainty strategy is discussed, especially with the MAN and the SDL-joint models for their good performance in Section 3.3.3.

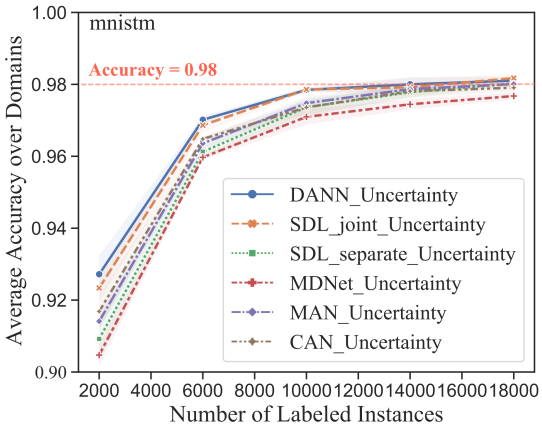
On the Digits dataset, the results of the Uncertainty strategy on both domains are plotted in Figure 3.4. The classification task on MNIST-M is supposed to be harder due to the more complex image backgrounds. The accuracy on MNIST-M is relatively lower than MNIST, as we expected. On MNIST-M, all the combinations obtain a near 98% accuracy when the budgets are depleted. However, on MNIST, the accuracy of 98% has been exceeded with less than 6000 overall labeled instances. Besides, the model-strategy combinations perform more concentrated on MNIST than on MNIST-M. These facts reflect the difficulty of the task on MNIST-M. SDL-joint with Uncertainty, which obtains the top overall performance on this dataset, achieves an above-average performance on the easier



(a) Digits (both domains)



(b) MNIST domain



(c) MNIST-M domain

Figure 3.4: The overall performance (a) and performance in each domain (b, c) from the Digits dataset. The lighter area represents the standard deviation.

MNIST domain and outperforms the others on the harder MNIST-M domain. Besides, MAN with Uncertainty performs worse on digit dataset compared to its performance on the rest five datasets. It can still obtain good performance on MNIST and medium performance on MNIST-M. The performances of different domains on rest of the datasets can be found in the appendix A.2.2. On these datasets, according to the learning curves, MAN with Uncertainty usually performs well in most domains, and it can at least obtain above-average performance in the worst performed domain.

In conclusion, the model-strategy combinations (such as MAN-Uncertainty & SDL-joint-Uncertainty) with the best overall performance usually cannot consistently perform

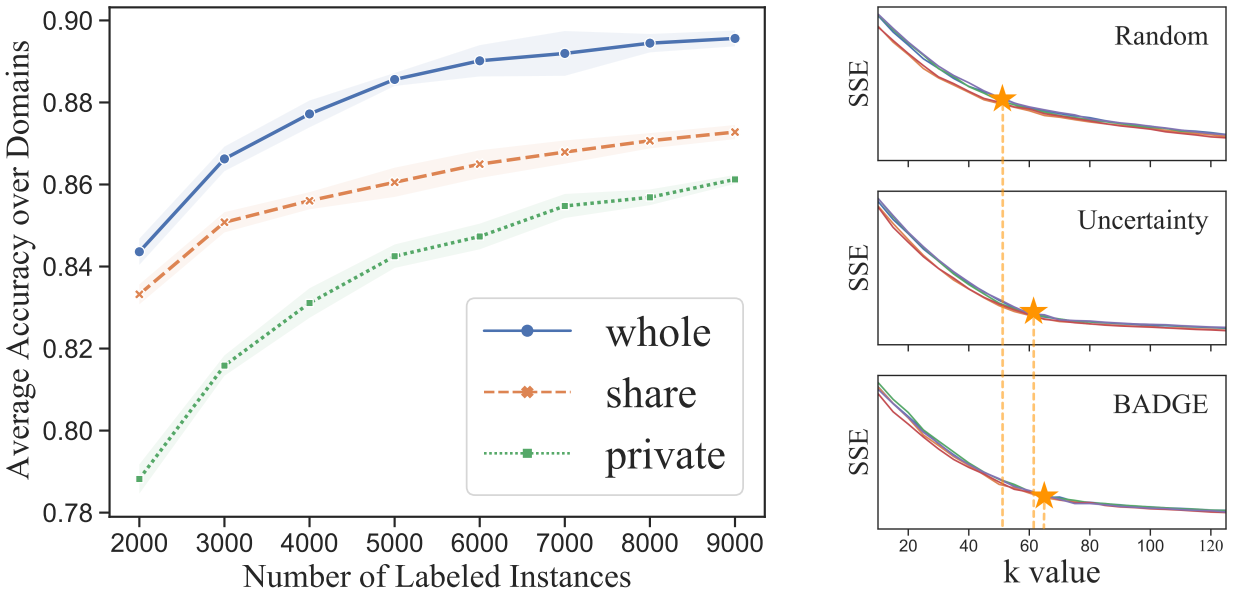
best on all the domains. However, they can at least obtain medium performance in the worst performed domain in the worse case. Hence, such combinations can be applied to real scenarios since domain performance can be ensured.

3.5 Deeper Investigations

In the previous section, we compared the models and strategies in the MDAL pipeline. The results indicated that the MAN and SDL-joint models, when coupled with the BvSB uncertainty strategy, demonstrated promising performance in most or particular datasets. Building upon these findings, this section will delve deeper into the reasons behind their superior performances as observed in the previous comparison.

The share-private structure guarantees the good overall performance of MAN. The shared feature extractor ensures good initial performance, and the domain-specific extractor captures the domain-specific information at the latter stage of the learning. An ablation study is made on the Office-Home dataset to verify the utilities of the share-private structure. The trained MAN model is separated into a shared part and a private part, and each part can make predictions independently. The test performances of both parts are shown in Figure 3.5(a). At the beginning, the shared part dominates the performance, and the performance discrepancy between the shared part and the whole model is slight. The discrepancy increases with more labeled training instances. The growth rate of the private part is higher than the shared part, and the performances of both parts get closer at the end of the AL process. The overall performance at the latter stage relies more on the private part than at the former stage since solely using the shared part cannot get comparable performance with the whole model.

Without pretrained models, joint training performs well in deep models on raw image datasets. SDL-joint and DANN obtain the best performances with deep networks on the Digits and the PACS datasets. When the model is deep, the information from different domains can be captured by one single inference structure as long as there



(a) MAN analysis

(b) Diversity analysis of selections

Figure 3.5: The analysis of MAN and Uncertainty for their superior performance. (a) The performance of the shared and private parts of the MAN model on the Office-Home dataset. (b) The elbow method is used to evaluate the diversity of the batch selection by different strategies. The k value is decided at the turning point (yellow star) of the SSE-k curve.

are no conflicts among domains. The single inference structure with relatively more labeled training instances (instead of separating training data for each domain) can easily learn a better feature extractor and a better classifier. Thus, SDL-joint outperforms other MDL models on the Digits dataset. However, when a good model initialization (pretrained feature extractor) is used in the PACS dataset, SDL-joint's performance improvement relative to MDL models disappears as shown in Figure 3.3(f).

Besides, SDL-joint cannot handle domain conflicts, where similar items on different domains might have different ground truth labels. For instance, "It is hot" is a positive review for a heater rather than a refrigerator in a sentiment classification task. In this case, SDL-joint is impossible to obtain correct predictions for both domains because the outputs will remain the same. However, domain conflicts are not common in image classification, which avoids the performance degradation in this comparison. In some multi-domain natural language processing tasks, deep MDL models consistently obtain better performances than SDL-joint (Chen and Cardie, 2018; Wu, Inkpen, et al., 2021).

The weakness of the Uncertainty strategy in redundant selection (low diversity) is mitigated in MDAL. The naive Uncertainty strategy surprisingly performs well in most cases, while it usually performs worse than other state-of-the-art strategies in the conventional single domain AL. The biggest weakness of Uncertainty is selecting similar instances in a single batch (Ash, Zhang, et al., 2020). We believe this weakness of Uncertainty is relieved by MDAL, where the existence of domains brings higher intra-batch diversity in uncertainty selections. The diversity of the selected instances by Random, Uncertainty, and BADGE strategy is analyzed. Considering the first selected batch, the inducted gradients for the last fully connected layer parameters are taken as the embeddings for the instances. In terms of the interpretability of neural networks (Zhang, Tiňo, et al., 2021), these gradients reveal the influences from different training instances to the model. The appropriate number of clusters k in k-means can be used to evaluate the diversity of the batch selection. The k is decided by the elbow method, which takes the k at the turning point of the loss-k curve, as shown in Figure 3.5(b). The loss term is the sum of square error (SSE) between

the instances and their cluster centers. BADGE is supposed to have the highest diversity on the gradient embedding. The k value of the Uncertainty selection is close to BADGE, which supports the diversity of the Uncertainty selection in MDAL.

3.6 Chapter Summary & Discussion

In this chapter, we present the first comprehensive comparative study on MDAL, offering a general pipeline as an off-the-shelf solution for MDAL. Within this pipeline, we compare the performance of six models from the MDL setting, combined with five AL strategies, across six datasets.

The comparison of models reveals the superiority of MAN with the shallow neural networks and the superiority of SDL-joint with the deep neural networks. Additionally, Uncertainty proves to be a competitive strategy compared to state-of-the-art alternatives in the AL strategy comparison. Specifically, when utilized with the MAN model, Uncertainty consistently achieves top performance across most datasets. Moreover, the combinations that yield strong overall performances also exhibit robustness in different domains, with their in-domain performances consistently ranking above average. Further investigation into their superiority reinforces our recommendation to employ MAN with Uncertainty in MDAL applications due to its exceptional performance in our experiments.

In the future, this work can be extended in the following directions:

- The evaluations of the conventional AL strategies in the pipeline are solely based on the outputs of MDL models. They are not optimal for the whole learning system since the instances most beneficial to the current model (for the current domain) might not bring the most improvement to the overall performance. A new ad-hoc MDAL strategy that can explicitly evaluate the domain-shared and domain-private informativeness should be developed.

- The current pipeline has not set importance weights to specific domains, while the

importance of distinct domains can be different in real life. It is interesting to update the current pipeline to handle the weights during the training and querying.

- Considering the newly added domains in real applications, it is interesting to generalize this pipeline to an active domain generalization problem or a multi-source active domain adaptation problem, where the performance on the new domains is concerned.

- The relations among domains are only revealed through the MDL models. It is interesting to include other correlation measurements in the pipeline.

Chapter 4

Multi-Domain Learning from Insufficient Annotations

In Chapter 3, we present the first comprehensive comparative study on multi-domain active learning (MDAL), encompassing a general MDAL pipeline and a series of comparisons among strategy-model pairs. The training of the MDL model is a crucial component of the entire MDAL pipeline. It involves simultaneously constructing a model or a set of models using datasets collected from different domains. Conventional approaches prioritize the extraction of domain-shared information and the preservation of domain-private information, adhering to the shared-private framework (SP models), which offers significant advantages over single-domain learning. However, most MDL models were designed for a relatively sufficient number of labeled instances under a supervised learning manner. This limitation hinders the effectiveness of conventional supervised MDL approaches in real-world applications, and also affect the performance of MDAL during the initial insufficient annotated stage. As a result, it becomes necessary to explore the MDL problem with insufficient annotations. Thus, a natural question arises: **how can we achieve cost-efficient MDL with limited annotated data?**

In this chapter, we present a novel method called multi-domain contrastive learning

(MDCL) designed to mitigate the impact of insufficient annotations by capturing both semantic and structural information from both labeled and unlabeled data. Specifically, MDCL consists of two key modules: inter-domain semantic alignment and intra-domain contrast. The former aims to align annotated instances of the same semantic category from distinct domains within a shared hidden space, while the latter focuses on learning a cluster structure of unlabeled instances in a private hidden space for each domain. Importantly, MDCL seamlessly integrates with many SP models, requiring no additional model parameters and enabling end-to-end training. Our experiments on five textual and image multi-domain datasets demonstrate that MDCL outperforms various SP models, leading to noticeable improvements. Furthermore, when incorporated into MDAL, MDCL contributes to a superior initialization, resulting in enhanced overall performance.

The remainder of this chapter is organized as follows. Section 4.1 provides the background and motivation of this chapter. In Section 4.2, we present the problem formulation of MDL with limited annotations. The methodology of MDCL is described in Section 4.3. Section 4.4 presents the experimental setup and results addressing the research questions. Finally, Section 4.5 offers concluding remarks for this chapter.

This chapter has been previously published in the Work 2 (He, Liu, Wu, et al., 2023). It should be emphasized that this version includes additional experimental runs and enhanced statistical analysis, offering a more thorough evaluation of the proposed method.

4.1 Background and Motivation

In many machine learning tasks, models are built on datasets collected from various data sources with different distributions, known as domains. For instance, texts from different sources like news articles, social media posts, and scientific papers constitute distinct domains in natural language processing. In computer vision, images of differing styles, such as sketches, cartoons, art paintings, and camera photos (Li, Yang, Song, et al., 2017) are considered distinct domains. While each domain possesses unique information, they often

share a significant amount of information with other domains. Naive solutions involve jointly building a single model across domains or independently constructing models for each domain, as is done in conventional single domain learning (SDL) approaches. However, joint training may neglect the unique information on each domain, while independent training disregards the correlations among domains (Liu, Qiu, et al., 2017). To address these shortcomings, multi-domain learning (MDL)(Dredze and Crammer, 2008) has been proposed to simultaneously capture domain-shared and domain-private information. Most existing MDL works concentrate on sharing information among domains while preserving domain-private information through models under the shared-private framework (SP models) (Liu, Qiu, et al., 2017; Chen and Cardie, 2018). Typically, following the concept of domain adaptation (DA) (Pan and Yang, 2010), shared information can be captured through distribution alignment across domains, allowing several DA methods to be utilized in MDL. Besides, private information is usually managed by a private component of the model for each domain. Accounting for both types of information has led to significant performance improvements over joint and independent training in the past decade (Liu, Qiu, et al., 2017).

In real-world applications, obtaining a sufficiently labeled dataset can be costly, even within a single domain (Zhan, Liu, et al., 2021; Tang, Liu, et al., 2021; Liu, Tang, et al., 2022). This issue exacerbated in MDL since constructing labeled multi-domain dataset is even more challenging due to the difficulty in accessing data from multiple domain experts (Huang, Han, et al., 2019; Zheng et al., 2021; Mghabbar and Ratnamogan, 2020). For instance, in the case of multi-domain medical image datasets (Huang, Han, et al., 2019), the high cost of manual annotations from medical experts across various research fields is just one of the challenges. The varying privacy and legal concerns, quality assurance processes, and labeling tools across domains also entail additional costs. The aforementioned MDL approaches face challenges in this high-cost scenario as they heavily rely on fully supervised training from a relatively sufficiently annotated multi-domain dataset. Therefore, a natural question arises: **can we perform cost-efficient MDL with insufficient annotated data?**

To the best of our knowledge, only a few works address the issue of insufficient annotations from multiple domains. Some works utilize contrastive (Tanwisuth et al., 2021; Singh, 2021) and semi-supervised (Li, Liu, et al., 2021) learning to alleviate the impact of insufficient annotations across domains. The key is to utilize unlabeled data to improve the performance. Although these studies are most close to our work technically, they focus on the domain adaptation problem, where only the target domain performance is concerned and employ a single feature extractor for both domains. They do not consider the MDL setting, where both shared and private information should be taken into account and further been handled by certain model components. Besides, there is no fully labeled source domain in our setting to extract reliable class prototypes as in (Tanwisuth et al., 2021; Singh, 2021; Li, Liu, et al., 2021). Other works propose to utilize active learning (AL) (Settles, 2009) in MDL, which is referred to as multi-domain active learning (MDAL) (He, Liu, He, et al., 2023). Given a budget for annotation, MDAL begins with a small set of labeled instances and iteratively selects the new instances for model building. However, without a good initial model trained on insufficient labeled instances, the selection process is likely to be biased and unreliable in the subsequent MDAL iterations. In summary, no method is readily applicable for MDL with insufficient annotations so far.

In this chapter, we propose a novel MDL approach, called multi-domain contrastive learning (MDCL), to construct neural network models on a limited number of labeled instances from each domain. Figure 4.1 presents an intuitive understanding of MDCL, where the semantic and structural information are respectively captured from labeled and unlabeled data. Specifically, MDCL comprises two components: a supervised contrastive loss to align instances of the same category from different domains within a shared hidden space and an unsupervised contrastive loss that focuses on learning the cluster structure of instances from the same domain in a private hidden space. By integrating both components, MDCL can learn a well-aligned representation from insufficient annotations. Importantly, MDCL is readily compatible with various share-private (SP) models for MDL (Bousmalis et al., 2016; Liu, Qiu, et al., 2017; Chen and Cardie, 2018), requiring no additional model parameters

and allowing for end-to-end training. Experimental results across five textual and image multi-domain datasets demonstrate that MDCL brings noticeable improvement over various SP models.

The main contributions of this chapter are summarized as follows:

- We introduce a novel approach called MDCL, designed to build neural network models on datasets with insufficient labels in the context of MDL. The MDCL approach seamlessly integrates with various SP models, requiring no additional model parameters and enabling end-to-end training.

- Through extensive experimentation, we demonstrate that MDCL outperforms many existing SP models across five textual and image multi-domain datasets. The results show a noticeable and consistent improvement in model performance, indicating the efficacy of our proposed approach.

- We show that MDCL can be effectively applied in the context of MDAL. By leveraging limited annotations, MDCL achieves superior model initialization, resulting in overall performance improvements for MDAL.

4.2 Problem Formulation

In Section 2.3.2, we have given the formal definition of multi-domain learning (MDL). However, in that definition, we assume that the labeled instances are sufficient for each domain. Thus, we re-formulate the problem of multi-domain learning, which further considers the insufficient annotations and utilizes both labeled and unlabeled instances. The formulation is written as follows:

Given K different domains (distributions) $\mathcal{D} = \{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(K)}\}$, a set of data pools $\mathcal{P} = \{\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(K)}\}$ containing both labeled and unlabeled data is collected from $\mathcal{D}$ in advance. The labeled data from each pool constitute a labeled data set $\mathcal{L} = \{\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \ldots, \mathcal{L}^{(K)}\}$, where $\mathcal{L}^{(k)} = \left\{(x_i^{(k)}, y_i^{(k)})\right\}_{i=1}^{|\mathcal{L}^{(k)}|}$ for $k = 1, 2, ..., K$. Considering the

scenario with limited labeled instances, the number of the labeled instances is much smaller than the collected data pool, i.e. $|\mathcal{L}^{(k)}| \ll |\mathcal{P}^{(k)}|$. MDL under insufficient annotations is is to find a set of models $F_\Theta = \{f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_K}\}$ for K domains by utilizing the common knowledge of different domains, which can be expressed as follows:

$$\Theta^*(\mathcal{L}, \mathcal{P}) = \arg\min_{\Theta} \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{|\mathcal{L}^{(k)}|} \sum_{i=1}^{|\mathcal{L}^{(k)}|} L(F_\Theta(x_i^{(k)}), y_i^{(k)}) \right) + \Omega(F_\Theta, \mathcal{P}) \tag{4.1}$$

where L is the loss function that measures the discrepancy between the predictions of the model and the ground truth labels. $\Omega(F_\Theta, \mathcal{P})$ denotes a general structural risk term on the model parameters Θ and data pools $\mathcal{P}$ for capturing the common knowledge.

This chapter is dedicated to the model training process, as described in Equation 4.1. The objective is to devise a novel model training approach that effectively leverages both a limited number of labeled instances and unlabeled instances, ultimately yielding a trained model F_Θ with a low generalization error. The key challenge is to capture the common knowledge across domains through the structural risk term $\Omega(F_\Theta, \mathcal{P})$ with limited labeled instances.

4.3 Methodology

This section introduces our novel multi-domain contrastive learning (MDCL) method for MDL with limited annotations. The key is to maximally utilize the unlabeled data pools and limited labeled instances to learn a set of models for K domains. The labeled data can be used to capture the semantic information and align the distributions across domains. The unlabeled data can further be used to learn the structural information of the data distribution and preserve the local information. MDCL introduces an inter-domain semantic alignment loss and an intra-domain representation learning loss to utilize the unlabeled and limited labeled data maximally. As we introduced in Section 2.5, the share-private framework is the commonly used architecture for MDL. MDCL takes the share-private framework as the

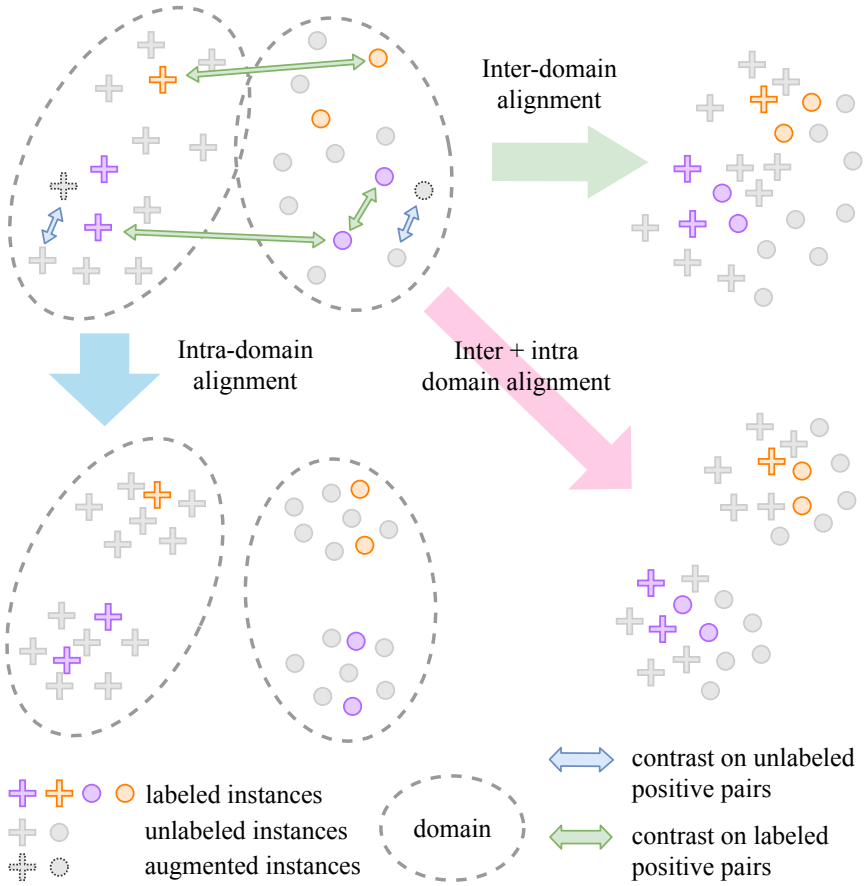


Figure 4.1: **Intuitive understanding of MDCL.** An illustrative example in the hidden space: Different colors represent different categories, and different shapes represent different domains. MDCL conducts two types of alignments to capture semantic and structural information from both labeled and unlabeled data: **Inter-domain alignment** aims to align items within the same category but from different domains closer to each other. **Intra-domain contrast** aims to maintain a cluster structure in each domain and make instances more separable.

backbone, where our designed loss can be easily integrated into the existing MDL methods as a plug-and-play component and trained in an end-to-end manner without introducing additional model parameters. The inter-domain semantic alignment and the intra-domain representation learning loss are based on the contrastive learning to guide the feature extraction.

The overview of MDCL is depicted in Fig. 4.1. Utilizing labeled instances in the inter-domain alignment ensures the shared semantic information is captured. Besides, amount of unlabeled instances in the intra-domain contrast ensures the structural information is preserved. With both types of information, the model can be trained to capture the common knowledge across domains effectively, thus leading to a better performance with less labeled instances. The remainder of this section will first provide an overview of the share-private framework, which serves as the backbone of MDCL, and then describe our method and its components in detail.

4.3.1 Share-Private Framework

The popular solution for MDL is the share-private framework (SP structure), which uses two types of feature extractors to handle both domain-shared and domain-private information. It effectively combines shared and private feature extractors to capture domain-shared and domain-specific information in MDL. The shared-information across domains could improve overall performance by training a shared extractor, while domain-specific information may be lost during the joint training. Whereas, the private extractor can efficiently preserve domain-specific information. The representations from both extractors are then concatenated for inference. As a representative and efficient SP model, the structure of MAN (Chen and Cardie, 2018) is illustrated in Fig. 4.2.

Given an input instance, the shared feature extractor and the private feature extractor are used to extract the shared and private representations, respectively. A discriminator is used to predict the domain of the shared representation, and the objective is to maximize

the domain prediction loss. The shared and private representations are concatenated and fed into the classifier to predict the label, and the objective is to minimize the classification loss.

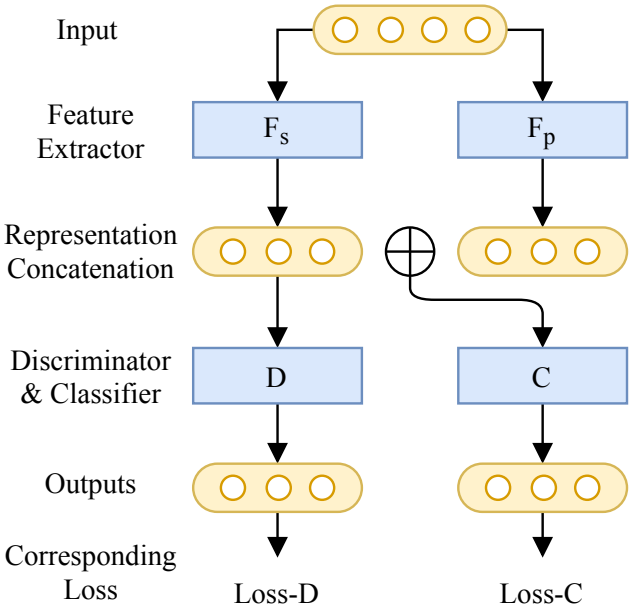


Figure 4.2: As a representative model for MDL under the share-private framework, MAN is taken as an illustration. The yellow parts represent vectors, and blue parts represent model components. The output vectors from the discriminator and the classifier are used to calculate the domain loss and the classification loss, respectively.

4.3.2 Inter-Domain Semantic Alignment

The conventional idea of MDL is to utilize a shared feature extractor (Ganin et al., 2016) to align the marginal distributions of domains. However, this approach may map instances within the same category far apart. Therefore, it is crucial to consider the conditional distribution as well (Motiian et al., 2017; Wu, Inkpen, et al., 2021), meaning that items with identical labels should be mapped closely in the latent space, which is referred to as inter-domain semantic alignment.

Different from the previous methods for DA (Tanwisuth et al., 2021; Singh, 2021; Li,

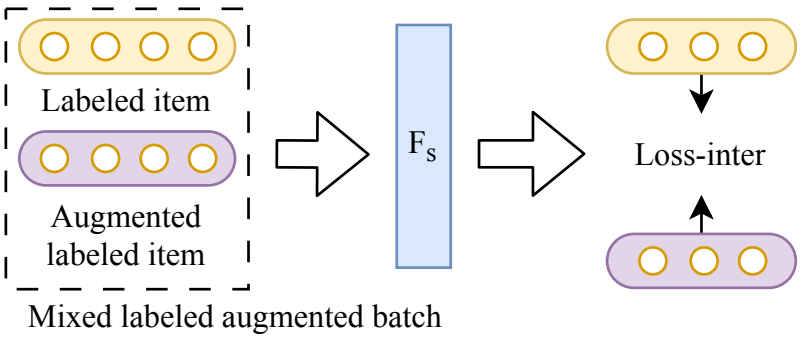


Figure 4.3: An illustration of the inter-domain semantic alignment process. Given the original item and its labeled augmentations, an inter-domain contrastive loss is applied on the outputs of the shared feature extractor to align the representations of items within the same category.

Liu, et al., 2021), in the scenario with limited labeled instances in MDL, obtaining reliable class prototypes is not feasible. In this situation, we could employ a NT-Xent contrastive loss (Chen, Kornblith, et al., 2020) in a supervised manner (Khosla et al., 2020) directly on the limited labels to proceed semantic alignment and bypass the construction of prototypes. Pairs of items should be mapped together as long as they belong to the same category, regardless of the domain they originate from. As illustrated in Fig. 4.3, we introduce an inter-domain semantic alignment loss on the domain-shared representation space, which can be expressed as:

$$L_{\text{inter}} = \sum_{i \in I} L_i = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau)} \tag{4.2}$$

where i is the index of a sample in the augmented batch $I \equiv \{1 \dots 2N\}$, and N is the batch size. $A(i) \equiv I \setminus \{i\}$. $P(i)$ is the index set of positives $\{p \in A(i) : \tilde{\boldsymbol{y}}_p = \tilde{\boldsymbol{y}}_i\}$, and $|P(i)|$ is the cardinality. $\boldsymbol{z}_l$ is the representation of the l-th item of the augmented batch extracted from the domain-shared feature extractor. τ is a scaler temperature factor. The original batch is chosen from a mixed labeled dataset that encompasses all domains, and then augmented to obtain I. This means that instances with the same label but from different domains can be

included in the same batch. By minimizing the inter-domain semantic alignment loss, the items within the same category from different domains could be aligned closer to each other in the shared representation space as they would have larger cosine similarity.

However, solely using the proposed inter-domain semantic alignment loss may not preserve the local manifold structure and only affect on the few labeled instances. As illustrated in Fig. 4.1, solely using inter-domain alignment might result in the unlabeled representations being mixed and losing the cluster structures.

4.3.3 Intra-Domain Representative Learning

Within each domain, unlabeled data can be effectively utilized for intra-domain contrastive alignment. This is based on the concept of instance contrastive learning (Singh, 2021). Unlike the previously employed inter-domain instance alignment, where intermediate representations were used, we now leverage classifier outputs to align items within each domain. This approach ensures that different augmentations, subjected to the same item, receives similar class assignments. By aligning the model outputs, this unsupervised training influences all the module parameters in SP models, including the shared feature extractor, private feature extractor, and classifier. Consequently, the contrastive alignment ensures a low density (uniformity) and form robust clusters (alignment) in the concatenated representation space (Wang and Isola, 2020). As a result, the domain representations and outputs are more robust to noise and augmentations. To achieve this unsupervised alignment, we also leverage the NT-Xent contrastive loss (Chen, Kornblith, et al., 2020). As illustrated in Fig. 4.4, we introduce an intra-domain contrastive loss on the domain-private output space, which can be expressed as:

$$L_{\text{Intra}} = \sum_{i \in I} L_i = -\sum_{i \in I} \log \frac{\exp(\boldsymbol{o}_i \cdot \boldsymbol{o}_p / \tau)}{\sum_{a \in A(i)} \exp(\boldsymbol{o}_i \cdot \boldsymbol{o}_a / \tau)} \tag{4.3}$$

where $\boldsymbol{o}$ is the output from the classifier, and $\boldsymbol{o}_p$ is the output of the corresponding positive instance, $\boldsymbol{o}_a$ is the output of the other instances in the augmented batch.

By minimizing the intra-domain contrastive loss, in each domain, the items originating from the same item could be aligned closer to each other in the output space of the classifier, as they would have larger cosine similarity. Consequently, the cluster structure of the domain-private representation space is preserved.

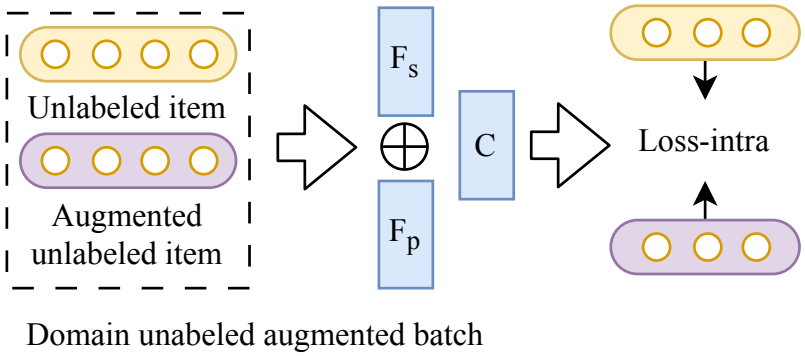


Figure 4.4: An illustration of the intra-domain representation learning process. Given the original item and the unlabeled augmentations, an intra-domain contrastive loss is applied on the outputs of the classifier to align the representations of items within the same domain.

However, solely using the proposed intra-domain contrastive loss only affects the alignment in each domain. As illustrated in Fig. 4.1, the items within same categories from different domains would still be far from each other in the representation space after solely using intra-domain alignment.

4.3.4 Overall Framework and Pseudocode

By incorporating inter-domain semantic alignment and intra-domain contrastive learning, MDCL demonstrates its ability to effectively leverage both limited labeled data and abundant unlabeled data for constructing MDL models. Notably, MDCL does not introduce any additional model parameters, allowing for seamless integration into various SP models. To provide a comprehensive overview of the training procedure, we present the pseudocode of MDCL in Algorithm 1, utilizing MAN (Chen and Cardie, 2018) as a backbone example to

facilitate better comprehension.

The inter-domain alignment is conducted first by aligning the shared feature representations of the labeled instances from different domains. Then, the discriminator is trained to discriminate the shared feature representations of the unlabeled instances from different domains. Finally, the main iteration is conducted, where the shared feature representations of the labeled instances are used to compute the classification loss, and the shared feature representations of the unlabeled instances are used to compute the domain loss and the intra-domain alignment loss. The domain prediction loss $L_{\mathcal{D}}$, $L^{\mathcal{D}}_{\mathcal{F}_s}$ and classification loss $L_{\mathcal{C}}$ are cross-entropy losses between the predictions and the ground truth. The entire process is repeated until convergence.

4.4 Experiments

4.4.1 Research Questions

Given the MDCL method, under the insufficient annotation scenario, the following research questions arise:

1. As a plug-and-play method, can MDCL enhance the performance of various models under the SP structure with the limited number of labeled instances (around 5%-20%) or extremely few labeled instances (around 1%)? (Section 4.4.3)

2. Since there are two technically independent components in MDCL, how does each of them affect the performance? (Section 4.4.4)

3. Given a further labeling budget, MDAL could be utilized. Can MDCL improve the entire MDAL process with a relatively large number of unlabeled instances (5%-50%)? (Section 4.4.5)

Algorithm 1 Multi-Domain Contrastive Learning.

Require: labeled dataset $\mathcal{L}$; unlabeled dataset $\mathcal{U}$; hyperparameter $\lambda_d > 0$, $\lambda_{inter} > 0$, $\lambda_{intra} > 0$, $k_{inter} \in \mathbb{N}, k_{adv} \in \mathbb{N}$

repeat
  ▷ Inter-domain alignment
  for $iter = 1$ to k_{inter} **do**
    $l_{inter} = 0$
    Sample a mini-batch $(\boldsymbol{x}, y) \sim \mathcal{L}$ ▷ From mixed labeled dataset
    $\boldsymbol{x\prime}, y\prime = \text{Aug}(\boldsymbol{x}, y)$ ▷ Augmented batch
    $\boldsymbol{z}_s = \mathcal{F}_s(\boldsymbol{x})$; $\boldsymbol{z}'_s = \mathcal{F}_s(\boldsymbol{x\prime})$ ▷ Shared feature vector
    $l_{inter} = \lambda_{inter} \cdot L_{inter}(\boldsymbol{z}_s, \boldsymbol{z}'_s; y, y\prime)$ ▷ **Inter-domain loss (Equ. 4.2)**
    Update $\mathcal{F}_s$ parameters using ∇l_{inter}
  ▷ Discriminator training
  for $iter = 1$ to k_{adv} **do**
    $l_{\mathcal{D}} = 0$
    for all $d \in \Delta$ **do** ▷ For all N domains
      Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_d$
      $\boldsymbol{z}_s = \mathcal{F}_s(\boldsymbol{x})$
      $l_{\mathcal{D}}$ += $L_{\mathcal{D}}(\mathcal{D}(\boldsymbol{z}_s); d)$ ▷ Accumulate $\mathcal{D}$ loss
    Update $\mathcal{D}$ parameters using $\nabla l_{\mathcal{D}}$
  ▷ Main iteration
  $loss = 0$
  for all $d \in \Delta_L$ **do** ▷ For all labeled domains
    Sample a mini-batch $(\boldsymbol{x}, y) \sim \mathcal{L}_d$
    $\boldsymbol{z}_s = \mathcal{F}_s(\boldsymbol{x})$
    $\boldsymbol{z}_d = \mathcal{F}_d(\boldsymbol{x})$ ▷ Domain feature vector
    $loss$ += $L_{\mathcal{C}}(\mathcal{C}(\boldsymbol{z}_s, \boldsymbol{z}_d); y)$ ▷ Compute $\mathcal{C}$ loss
  for all $d \in \Delta$ **do** ▷ For all N domains
    Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_d$
    ▷ Discriminate
    $\boldsymbol{z}_s = \mathcal{F}_s(\boldsymbol{x})$
    $loss$ += $\lambda_d \cdot L^{\mathcal{D}}_{\mathcal{F}_s}(\mathcal{D}(\boldsymbol{z}_s); d)$ ▷ Domain loss of $\mathcal{F}_s$
    ▷ Intra-domain alignment
    $\boldsymbol{x\prime} = \text{Aug}(\boldsymbol{x})$ ▷ Augmented batch
    $\boldsymbol{o} = \mathcal{C}(\mathcal{F}_s(\boldsymbol{x}, \mathcal{F}_d(\boldsymbol{x}))); \boldsymbol{o\prime} = \mathcal{C}(\mathcal{F}_s(\boldsymbol{x\prime}, \mathcal{F}_d(\boldsymbol{x\prime}))$; ▷ Output of the classifier
    $loss$ += $\lambda_{intra} \cdot L_{intra}(\boldsymbol{o}, \boldsymbol{o\prime})$ ▷ **Intra-domain loss (Equ. 4.3)**
  Update $\mathcal{F}_s$, $\mathcal{F}_d$, $\mathcal{C}$ parameters using $\nabla loss$
until convergence

4.4.2 Experimental Setup

4.4.2.1 Dataset

We evaluate our proposed MDCL method on five popular multi-domain textual and image datasets, namely Amazon (Chen, Xu, et al., 2012), MNIST-USPS (Tzeng et al., 2017), Office-Home (Venkateswara et al., 2017), FDUMTL (Liu, Qiu, et al., 2017), and PACS (Li, Yang, Song, et al., 2017). These datasets at least contain two domains. All the tasks are classification tasks, and the categories are the same across domains. The number of instances and the train-validation-test partitions are also presented.

- **Amazon** dataset consists of four textual domains: books, dvd, electronics, and kitchen. Each domain contains two categories, with instances encoded to a vector representation of length 5000. There are 2000 samples on each domain. The ratio 6:2:2 is used to split the training/validation/test sets.

- **MNIST-USPS** contains two domains (sub-datasets): MNIST (LeCun, Bottou, et al., 1998) and USPS (Hull, 1994). Each domain comprises two image domains with ten categories each, and instances are encoded to a vector representation of length 256. There are 1000 samples on each domain. The ratio 6:2:2 is used to split the training/validation/test sets.

- **Office-Home** contains four distinct domains: art, clip-art, product, and real-world. Each of the four domains has sixty-five categories, with instances encoded to a vector representation of length 2048. There are 2427/4365/4439/4357 samples on domain 'Art', 'Clipart', 'Product', and 'RealWorld', respectively. The ratio 6:2:2 is used to split the training/validation/test sets.

- **FDUMTL** consists of sixteen textual domains, each containing two categories, and utilizes word2vec embedding for raw texts. In addition to using all the domains, we also conduct evaluations on the first four domains in our experiment, referred to as

FDUMTL (4 domains). There are 2000 samples on each domain. The ratio 6:2:2 is used to split the training/validation/test sets.

- **PACS** is an image dataset that contains seven categories from four domains: art-painting, cartoon, photo, and sketch. The split can be found in the following link[1]. There are 1840/2107/1499/3531 instances on the training set, 208/237/171/398 instances on the validation set, and 2048/2344/1670/3929 instances on the test set for each domain, respectively. Each instance is an RGB image with dimension (227,227,3).

4.4.2.2 Models

Our primary focus is evaluating the performance of MDCL on a single model. Therefore, we selected the most renowned and widely recognized models in MDL as baselines. In the majority of the experiments, MAN (Chen and Cardie, 2018) is utilized as the backbone of MDCL due to its simple structure and wide acceptance in the literature. ASP-MTL (Liu, Qiu, et al., 2017), as the most classic MDL model, is also used to verify the generation ability of MDCL. Compared to MAN, ASP-MTL has specific classifiers for each domain, while MAN has a shared classifier for all domains. While there are newer models available, such as CAN (Wu, Inkpen, et al., 2021), the existing literature (He, Liu, He, et al., 2023) suggests that these models do not outperform MAN. Some other recently proposed models in the MDL are designed for particular tasks (Mghabbar and Ratnamogan, 2020; Huang, Han, et al., 2019), which are beyond the scope of our paper.

The balancing parameter λ_d for $l_{\mathcal{D}}$ is set to 0.05 for all the datasets. For Amazon, MNIST-USPS and Office-Home datasets, we use one fully connected layer with a sigmoid activation function as the $\mathcal{F}_s$ and $\mathcal{F}_d$ feature extractor, both with the same output size of 64. For the FDUMTL dataset, we use a CNN as the feature extractor that takes 100d word embeddings as input from the sequence, and the output representation length is 128. A

[1] https://drive.google.com/drive/folders/0B6x7gtvErXgfUU1WcGY5SzdwZVk?resourcekey=0-2fvpQY_QSyJf2uIECzqPuQ

single convolution layer with 200 kernels of sizes 3, 4, and 5 is used. For the PACS dataset, we use a pre-trained Resnet-18 (He, Zhang, et al., 2016) as the feature extractor with an output size of 64. For all the datasets, the classifier and domain discriminator consist of a single fully connected layer. Table 4.1 provides all the hyper-parameters for optimizations in training details for each dataset.

4.4.2.3 MDCL Implementation Details

Table 4.1: The hyperparameters used for MDCL.

Datasets	Optimizer	Learning Rate	Learning Rate Decay	Batch Size	Weight Decay	Early Stopping	Inter-Domain λ & τ	Intra-Domain λ & τ
Amazon	Adam	3e-4	False	8	0.05	20	1/0.1	1/0.01
MNIST-USPS	Adam	3e-3	0.33	8	0.001	30	0.1/0.1	1/0.1
Office-Home	Adam	1e-2	0.33	8	0.001	15	1/0.01	1/0.01
FDUMTL	Adam	3e-4	0.1	8	0.001	30	0.1/0.01	1/0.01
PACS	SGD	1e-3	0.1	8	0.001	15	0.1/1	1/0.1

For MDCL, the inter-domain and intra-domain contrastive learning requires a balancing (λ) and a temperature (τ) hyperparameter, which are listed in Table 4.1. Given the overlap in models and datasets with the comparative study in Chapter 3, we predominantly adopted the hyperparameters from the previous work. For hyperparameters not previously encountered (e.g., setting a trade-off parameter and temperatures), we did engage in grid searching on validation set. Due to the computational complexity of batch augmentation, we set the batch size to 8 for all the datasets. Learning rate decay was applied to the optimizers to ensure the stability of the training process. Additionally, the early stopping technique was implemented during training to mitigate the risk of overfitting.

The augmentation batch generalization plays a crucial role in MDCL. Two types of augmentation techniques were employed in our study: 1) dropout (Hinton, Srivastava, et al., 2012) and 2) Gaussian noise. For the FDUMTL dataset, we utilized raw sentence input and sampled multiple outputs from dropout layers to achieve batch augmentation. As for the remaining datasets, where each instance is represented by a vector, we applied batch

augmentation using a Gaussian noise add-on with a standard deviation of 0.01. To ensure the reliability of our results, all experiments were conducted five times, and we calculated the average performance and standard deviation.

Same to the previous chapter, we utilized the accuracy as the primary performance metric for all the experiments. Learning curves were also plotted to visualize the performance of MDCL with different numbers of labeled instances. The area under the learning curve (AULC) was calculated for each curve. All the results are analyzed using the Mann-Whitney U test (Mann and Whitney, 1947), and the p-value is calculated to determine the statistical significance.

4.4.3 RQ1: Performance with Insufficient Labels

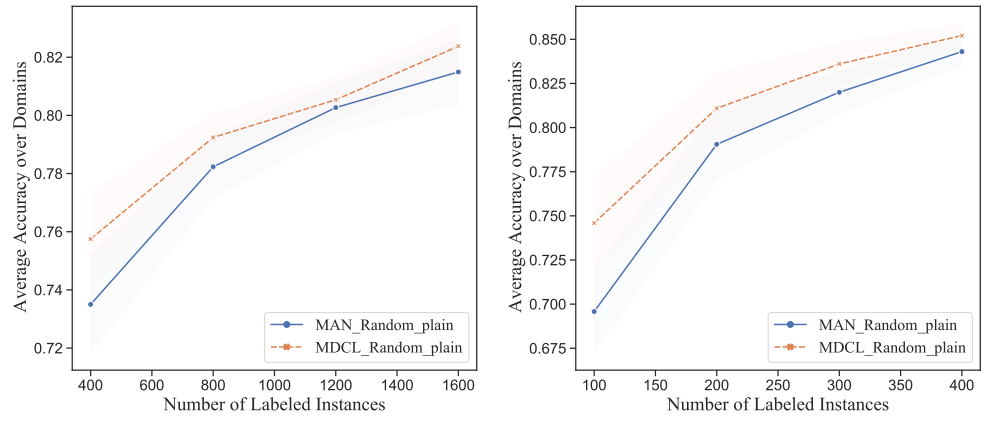
We evaluate the effectiveness of our MDCL method using a constrained number of labeled instances. Unlike few-shot learning (Wang, Yao, et al., 2021) in existing literature, where only a certain number of labeled instances are available for each category, we adopt a more practical approach by directly sampling a limited portion of labeled instances from the entire dataset for real-world applications. To address this research question, we consider two scenarios:

1. When the number of labeled instances is moderately limited (approximately 5%-20%),
2. When the number of labeled instances is extremely low (1%).

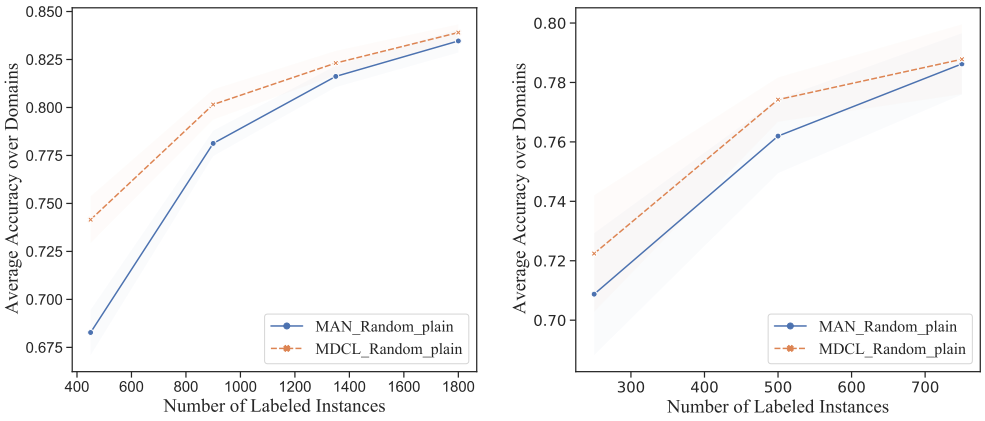
Both scenarios are assessed using the MAN model. Furthermore, we employ ASPMTL to verify the generation capability of MDCL over SP models in the case of limited labeled instances (around 5%).

4.4.3.1 Moderately Insufficient Labels

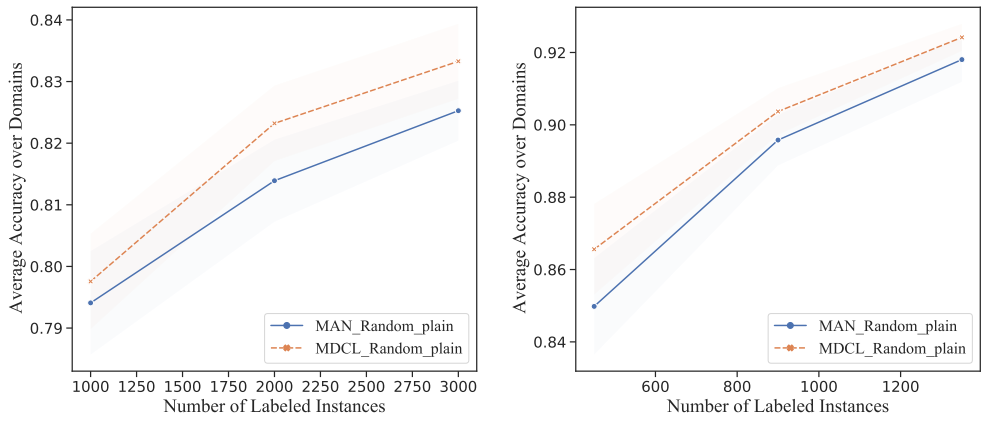
We assessed the efficacy of our method using various datasets. For the Amazon, MNIST-USPS, and Office-Home datasets, we utilized datasets with 5% to 20% labeled instances.



(a) Amazon (b) MNIST-USPS



(c) Office-Home (d) FDUMTL (4 domains)



(e) FDUMTL (f) PACS

Figure 4.5: The results of MDCL with different number of labeled instances on different datasets. The lighter area represents the standard deviation.

However, due to their high computational complexity, we employed datasets FDUMTL and PACS with 5% to 15% labeled instances.

The performance results are presented as learning curves, showcasing the outcomes with different numbers of labeled instances, as depicted in Figure 4.5. Across all the datasets, our MDCL approach consistently demonstrated significant improvements compared to using only MAN. Notably, the improvements were more pronounced when the number of labeled instances was limited. While the superiority of our method diminished as more labeled instances became available, MDCL still outperform the baseline model. The statistical significance of the results was confirmed, with the p-values presented in Table B.1.

4.4.3.2 Extremely Insufficient Labels

We conducted an evaluation of the performance of MDCL using an extremely small number of labeled instances on the Amazon, FDUMTL, and PACS datasets. Specifically, we trained the model using only 1% of the labeled instances available in each dataset. Since we randomly sampled the labeled instances, when the number of labeled instances is extremely low, the performance of the model may vary significantly. Thus, we conducted more than 20 runs for each experiment to ensure the reliability of the results. The detailed results can be found in Table 4.2.

Remarkably, significant improvements were observed across all the datasets. Notably, the most substantial enhancement was achieved on the Office-Home dataset, which contains a higher number of categories compared to the other two datasets. This outcome underscores the efficacy of MDCL, particularly in scenarios with limited labeled data, making it a promising approach for a wide range of applications. Although the deviations in the results were relatively high, the statistical significance of the results was confirmed, with the p-values less than 0.05.

Table 4.2: MDCL on only 1% labeled instances. Average performance in more than 20 runs with the standard deviation in parentheses.

Method	Amazon	MNIST-USPS	Office-Home	FDUMTL	PACS
MAN	0.6307	0.3922	0.3522	0.5335	0.6757
	(0.0191)	(0.0751)	(0.0247)	(0.0367)	(0.0234)
MDCL	0.6476	0.4190	0.4313	0.5673	0.6907
(+MAN)	(0.0273)	(0.1103)	(0.0289)	(0.0416)	(0.0202)
p-value	0.011	0.005	0.000	0.021	0.041

Table 4.3: MDCL with ASPMTL on 1% & 5% labeled instances. Average performance in 10 runs with the standard deviation in parentheses.

Method	Amazon		Office-Home	
	1%	5%	1%	5%
ASPMTL	0.5870	0.6874	0.1667	0.5816
	(0.0188)	(0.0128)	(0.0133)	(0.0152)
MDCL	0.6109	0.7224	0.2013	0.6141
(+ASPMTL)	(0.0162)	(0.0221)	(0.0399)	(0.0119)
p-value	0.014	0.004	0.031	0.001

4.4.3.3 Compatibility for Share-Private Framework

As a plug-and-play method, MDCL should be compatible with various SP (share-private) models, not limited to MAN. To evaluate the integration capability of MDCL with other models, we compare its performance with ASPMTL (Liu, Qiu, et al., 2017), a popular share-private framework model that has been widely used in the literature. We conduct our evaluation on the Amazon and Office-Home datasets, considering both 1% and 5% labeled instances. The results are presented in Table 4.3. Notably, MDCL exhibits superior performance compared to ASPMTL on both datasets, showcasing its strong generalization ability with different SP models. Overall, these findings demonstrate the compatibility and effectiveness of MDCL as a plug-and-play approach within diverse SP model architectures.

4.4.4 RQ2: Effectiveness of Components

In this section, we explore the effectiveness of the two components of MDCL, i.e., the inter-domain and the intra-domain contrastive loss. As two plug-and-play components, each component could be used independently. In this case, an ablation studies are conducted on the PACS and MNIST-USPS datasets with 5% labeled instances. The results are shown in Table 4.4. The statistical analysis of the results is presented in Table B.2 and Table B.3 in Appendix B.

The inter-domain contrastive proves more effective than the intra-domain contrast on PACS, but the opposite is observed for MNIST-USPS. In the statistical tests, we also observed superiority only in one of the components, with different components showing significance on different datasets. This discrepancy may be attributed to the fact that, on PACS, the pre-trained feature extractor already provides a reliable domain-invariant feature representation, making semantic information more critical. In contrast, on MNIST-USPS, the intra-domain contrastive loss is more effective, as it assists the feature extractor in learning features in each domain. As we summarized in Section 4.3.2 and Section 4.3.3, both components are designed to address the limitations of the other. The inter-domain semantic

alignment loss is designed to preserve the local manifold structure, while the intra-domain contrastive loss is designed to preserve the cluster structure for semantic alignment.

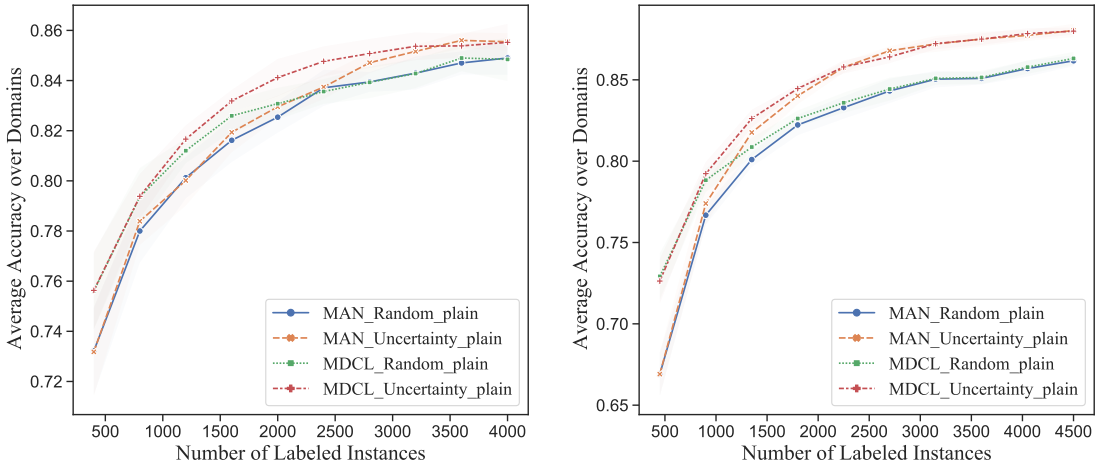
Table 4.4: Ablation study on PACS and MNIST-USPS datasets.

Method	MNIST-USPS	PACS
MAN	0.6982	0.8520
MDCL *(Inter)*	0.6989	0.8701
MDCL *(Intra)*	0.7465	0.8568
MDCL	0.7517	0.8730

4.4.5 RQ3: Ability to Integrate with MDAL

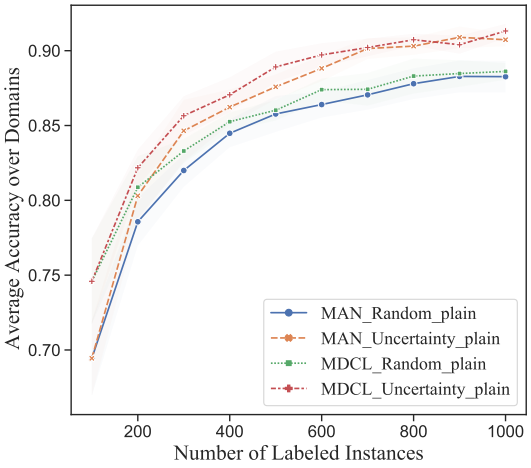
In Section 4.4.3, the learning curves indicate that the improvement decreases as more labeled instances are used. However, in real-world applications, a method that performs well only with fewer labeled instances but poorly with more labeled instances is not desirable. This is because the availability of labeled data might increase over time. To address this concern, it is crucial to evaluate the proposed method with a larger budget for collecting labeled instances, which allows for more labeled data to be included in the learning process. We envision a setting where the number of labeled instances can be expanded. This process of increasing the number of labeled instances in MDL is akin to multi-domain active learning (MDAL) (He, Liu, He, et al., 2023). In MDAL, informative instances are collected and added to the labeled dataset in each iteration.

Thus, we discover the potential of MDCL in MDAL setting, where the MDCL method is integrated with random selection and AL selection strategy. Specifically, we employ the MDAL strategy to iteratively collect informative labeled instances, with MDCL being trained in each iteration. For our experiments, we utilize the Amazon, MNIST-USPS, and Office-Home datasets and use the simplest yet effective AL query strategy, Best-vs-Second-Best (BvSB) (Joshi et al., 2009), for selecting instances. We vary the percentage of labeled instances from 5% to 50% and present the learning curves for all the datasets in Figure 4.6.



(a) Amazon

(b) Office-Home



(c) MNIST-USPS

Figure 4.6: The results of MDCL combined with the Uncertainty strategy in a MDAL setting.

Table 4.5: AULC of MDCL. Average performance in 10 runs with the standard deviation in parentheses.

Method	Amazon	MNIST -USPS	Office -Home
MAN	82.00	84.19	83.29
+Random	(0.41)	(0.85)	(0.21)
MDCL	82.43	85.26	84.03
+Random	(0.41)	(0.79)	(0.19)
MAN	82.43	86.43	85.03
+BvSB	(0.34)	(0.57)	(0.32)
MDCL	83.28	87.39	85.74
+BvSB	(0.31)	(0.40)	(0.18)

Moreover, we report the area-under-learning-curves (AULC) results for all three datasets in Table 4.5.

After analyzing the results, it is evident that the proposed MDCL method performs effectively with a limited number of labeled instances and remains competitive with MAN when a larger number of labeled instances are available. Moreover, when integrated with MDAL, MDCL demonstrates further performance improvements. The incorporation of BvSB leads MDCL to start from a more favorable initialization, resulting in more reliable selections for active learning. The learning curves in Figure 4.6 clearly illustrate how MDCL dominates MAN in both active or passive settings, throughout the entire learning process. Additionally, Table 4.5 presents evidence of MDCL's superiority, achieving higher AULC scores on all datasets.

These findings suggest that the MDCL method, particularly when combined with MDAL and BvSB, offers notable advantages in various learning scenarios. The results underscore the potential of MDCL as an effective and reliable approach in the context of active learning.

4.5 Chapter Summary & Discussion

In conclusion, we introduce a novel multi-domain contrastive learning (MDCL) approach for multi-domain learning. The primary objective of MDCL is to build neural network models on insufficient labeled instances from multiple domains. MDCL comprises two components: a supervised contrastive loss for inter-domain semantic alignment and an unsupervised contrastive loss for intra-domain representation learning. MDCL is readily compatible with many SP models, requiring no additional model parameters and allowing for end-to-end training. Experimental results across five textual and image multi-domain datasets demonstrate that MDCL brings noticeable improvement over various SP models. Additionally, given a labeling budget, MDCL can be further employed in multi-domain active learning to enhance the performance of the entire learning process.

In the future, this work can be extended in the following directions:

- **Multi-domain few-shot learning:** Currently, this work explores the setting of having the same proportion of labeled instances in each domain. However, this approach may result in situations where certain categories in certain domains have no labeled instances, i.e. 1% annotated setting. To address this, future research could focus on ensuring a minimum number of labeled instances in each category in each domain, leading to a multi-domain few-shot learning setting.

- **Multi-domain active learning:** Another promising direction is to apply MDCL in multi-domain active learning, as we introduced in Section 4.4.5. To achieve this, future studies could develop a unified framework that combines MDCL with a specific ad hoc active learning strategy. This strategy should not only select the most informative instances but also ensure that the selected instances align with the mechanism of MDCL.

- **Scalability of MDL:** While the current study demonstrates the effectiveness of MDCL across multiple domains with limited labeled data, there is room to investigate its scala-

bility for larger-scale data. Handling the high computational complexity of contrastive learning for big data poses a challenge. Therefore, future research could focus on optimizing algorithms for large-scale contrastive learning, which would significantly broaden the application of MDCL.

Chapter 5

Perturbation-Based Two-Stage Multi-Domain Active Learning

In Chapter 3, we emphasized the significance of both the model and strategy in multi-domain active learning (MDAL). However, as we highlighted in Section 2.6, there has been limited research conducted on multi-domain active learning (MDAL). Therefore, in Chapter 4, we delve into MDAL from the model perspective, where the focus lies on training the model with insufficient annotations, since the initialization of AL always faces this challenge. While the model perspective is essential, the strategy perspective is equally crucial in MDAL, yet it has received less attention in the existing literature. Previous studies have mostly relied on conventional active learning (AL) strategies for MDL scenarios, which fail to fully exploit the domain-shared information of each instance during the selection procedure. Addressing this gap, there is a need for an ad hoc strategy that takes cross-domain information into account. This naturally leads to the question: **how can we design effective AL strategies tailored explicitly for MDL?**

In this chapter, we introduce a novel perturbation-based two-stage multi-domain active learning (P2S-MDAL) method, which is incorporated into the well-regarded ASP-MTL model. Our P2S-MDAL approach involves the allocation of budgets for domains and the

establishment of regions for diversity selection as the first step. These regions are then used to identify the most cross-domain influential samples within each region. We introduce a perturbation metric to evaluate the robustness of the shared feature extractor of the model, which facilitates the identification of potentially cross-domain influential samples. To validate the effectiveness of our strategy, we conduct experiments on three real-world datasets, encompassing both texts and images, demonstrating superior performance compared to conventional AL strategies. Additionally, we carry out an ablation study to demonstrate the validity of each component. Finally, we outline several intriguing potential directions for future MDAL research, thus catalyzing advancements in the field.

The remainder of this chapter is organized as follows: In Section 5.1, we provide more details about the background and motivation behind this work. Section 5.2 presents the problem formulation. Next, in Section 5.3, we present a detailed explanation of the proposed method. The experimental settings are introduced in Section 5.4, and the results of the comparative study are presented in Section 5.5. Finally, in Section 5.6, we discuss the results and conclude this chapter.

This chapter is in Work 3 (He, Dai, et al., 2023) (Submitted to *CIKM-2023*). It should be emphasized that this version includes additional experimental runs and enhanced statistical analysis, offering a more thorough evaluation of the proposed method.

5.1 Background and Motivation

In practical applications, aggregating data from diverse sources is a common practice for accomplishing specific tasks. These data sources, often called "domains," exhibit distinct distributions. For instance, sentiment analysis may involve data collection from various social media platforms such as Twitter, Facebook, and Weibo. In image classification, different styles of images (Li, Yang, Song, et al., 2017), including sketches, cartoons, art paintings, and camera photos, represent distinct domains. Each domain possesses unique characteristics and contexts while containing sharable intertwined information. Although building separate

models for each domain is feasible, it fails to exploit the potential benefits of shared information. If appropriately harnessed, this shared information could significantly improve the performance of machine learning models. Multi-domain learning (MDL) (Dredze and Crammer, 2008) aims to simultaneously learn across various domains, leveraging shared knowledge to enhance overall performance. Empirically, MDL outperforms both single-domain learning and joint learning across multiple domains (He, Liu, He, et al., 2023) in many real-world applications.

However, high labeling effort represents a challenge in MDL, since data need to be collected from multiple domain experts. For example, in the context of multi-domain medical image classification, the manual annotation cost associated with different field experts is just one of the challenges (Huang, Han, et al., 2019). Additionally, varying legal, privacy, and ethical requirements and annotation tools across domains make the multi-domain data collection process more arduous. Therefore, **it is crucial to minimize the labeling effort in MDL.**

Active learning (AL) (Zhan, Liu, et al., 2021) presents a promising solution for reducing the labeling effort in machine learning tasks. By iteratively selecting informative samples for annotation, AL achieves comparable performance to random selection while requiring significantly less labeling effort. Several studies have explored the application of AL to reduce labeling effort in MDL, known as multi-domain active learning (**MDAL**) (Li, Jin, et al., 2012). Most works simply adapt conventional single-domain AL strategies to MDL models (He, Liu, He, et al., 2023; Li, Jin, et al., 2012), which leads to noticeable improvements. They mix all the evaluations from different domains and select the ones with the highest evaluation score. Nonetheless, solely applying single-domain AL to MDL models is suboptimal, as the domain-shared information remains underutilized in item selection. Besides, the mixed scores from different domains are incomparable, potentially leading to biased selection. To the best of our knowledge, no existing work has designed AL strategies specifically for MDL to tackle these issues. Thus, a natural question arises: **how can we design effective AL strategies tailored explicitly for MDL?**

This chapter proposes a novel perturbation-based two-stage multi-domain active learning (P2S-MDAL) strategy, which builds upon the classical and renowned MDL model, ASP-MTL (Liu, Qiu, et al., 2017). In the first stage, we allocate a budget to each domain to ensure a fair in-domain comparison and establish regions for diversity selection. Subsequently, within each region, we further select the most cross-domain influential samples for annotation. The influence evaluation is based on perturbations, a novel metric that assesses the robustness of the shared feature extractor of ASP-MTL. The underlying intuition is that if a sample is more informative to the shared extractor, it will be more vulnerable to perturbations, i.e., the perturbed sample will likely have more distinct outputs compared to the original one. Consequently, such examples, which are less learned by the shared feature extractor, could be more influential to all the domains. Experimental results on three real-world datasets, encompassing texts and images, demonstrate the superiority of the proposed P2S-MDAL strategy over conventional AL strategies.

The main contributions of this chapter are summarized as follows:

- We present a novel AL strategy named P2S-MDAL, specifically designed for the MDL scenario, built upon the renowned ASP-MTL model. Through extensive experimentation, we demonstrate the top-tier performance of P2S-MDAL in the comparison.
- We incorporate perturbations to assess the cross-domain influences of instances in AL. This approach offers a fresh and insightful perspective for evaluating the potential of individual instances.
- Our study sheds light on several intriguing research directions in the field of MDAL, opening up new possibilities for future exploration.

5.2 Problem Formulation

In Section 2.3.3, we have given the formal definition of MDAL from both bilevel optimization perspective and sequential selection perspective. This chapter follows the sequential selection

perspective of MDAL, which is defined in Definition 8. Specifically, in the i-th AL iteration, a batch of to-be-queried instances $\mathcal{Q}_i$ is selected from unlabeled data pool $\mathcal{U}_{i-1}$ according to the selection criteria α, and then annotated by an oracle:

$$\mathcal{Q}_i = \alpha(F_{\Theta^*(\mathcal{L}_{i-1},\mathcal{P})}, \mathcal{U}_{i-1}), \quad where \quad \mathcal{Q}_i \subseteq \mathcal{U}_{i-1}, |\mathcal{Q}_i| = b \tag{5.1}$$

$\mathcal{L}_{i-1}$ and $\mathcal{U}_{i-1}$ are then updated with the selected batch $\mathcal{Q}_i$, i.e., $\mathcal{L}_i = \mathcal{L}_{i-1} \cup \mathcal{Q}_i$ and $\mathcal{U}_i = \mathcal{U}_{i-1} \backslash \mathcal{Q}_i$. In the meantime, the model $F_{\Theta^*(\mathcal{L}_i,\mathcal{P})}$ is trained on the updated data $\mathcal{L}_i$ and data pool $\mathcal{P}$ with the following objective.

The objective of this chapter is to devise a selection criterion α that effectively chooses the most informative instance set $\mathcal{L}$ from the unlabeled set $\mathcal{U}$. In Equation 5.1, the selection criterion α is a function that directly yields the selected instance set $\mathcal{Q}_i$ from the input data $\mathcal{U}_{i-1}$ with the assistance of the model $F_{\Theta^*(\mathcal{L}_{i-1},\mathcal{P})}$.

Within the broader context of general active learning, this selection function α can typically be formalized in various ways, thereby differing among different active learning strategies. This function may include a scoring criterion for each instance in $\mathcal{U}_{i-1}$, where the top-b instances with the highest scores are selected, known as the score-based method. Alternatively, this function may not involve explicit score calculation, but rather the instances are chosen based on the distribution of the output groups, referred to as the representative-impart method. Consequently, within the realm of multi-domain active learning, the task of constructing a selection function essentially entails designing a scoring criterion, a representative criterion, or a combination of both.

5.3 Methodology

To address the limitations of conventional AL strategies, we propose a novel method P2S-MDAL for MDAL that evaluates the influence of samples on other domains. P2S-MDAL follows a two-stage framework: selecting regions establishment and perturbation-based item

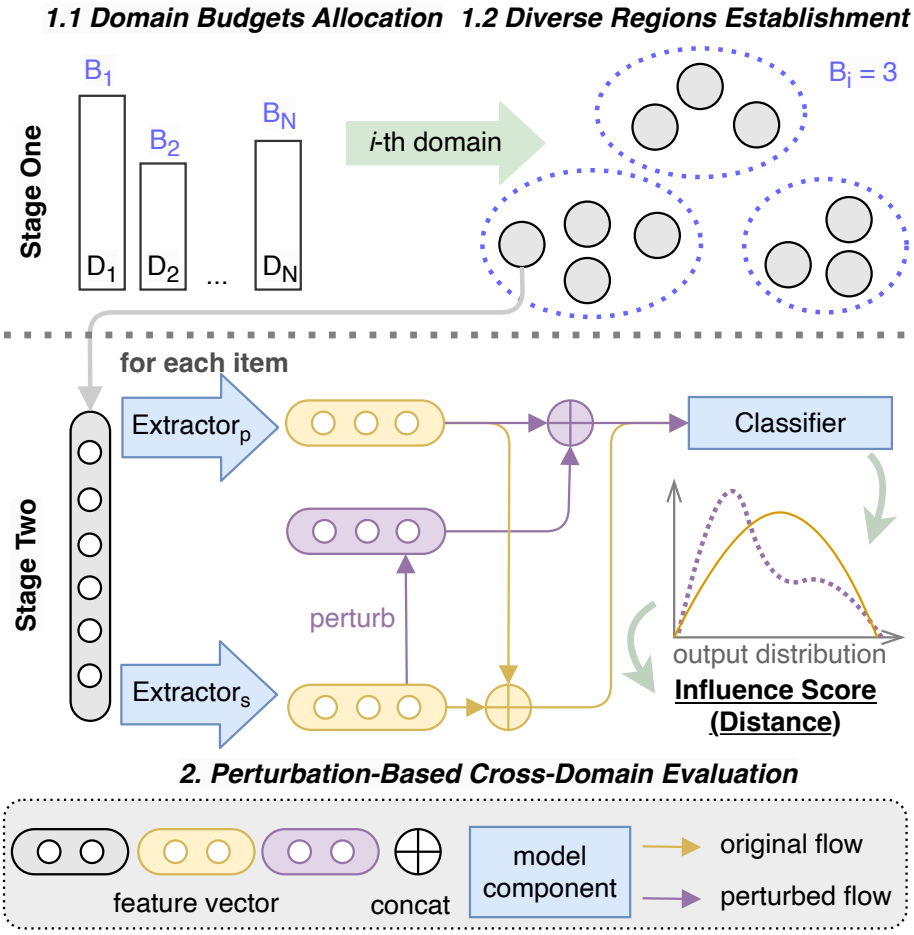


Figure 5.1: An illustration of the proposed P2S-MDAL method. First, selecting regions are established by a budget allocation and a selection space division processes. Then, samples with higher cross-domain influence score would be selected from each region.

evaluation, which are introduced in Section 5.3.1 and Section 5.3.2 respectively. The overall framework is illustrated in Figure 5.1. Several regions are pre-established from each domain in the first stage to ensure the diversity of the sample selection. In the second stage, the cross-domain influence of samples in each region is evaluated, ensuring that the selected samples benefit not only the current domain but also other domains. The pseudocode of the proposed method is shown in Section 5.3.3.

5.3.1 Stage One: Selecting Regions Establishment

To avoid the incompatibility between sample evaluation scores from different domains, we constrain the score-based selection within each domain. Thus, the budgets should be allocated to each domain in advance according to the influence of domains. Here we take the total number of samples as an influence estimate. Let n_k denote the number of samples in domain k, and B denote the total budget. The budget allocation is calculated as follows:

$$B_k = \frac{n_k}{\sum_{i=1}^{K} n_i} \times B \tag{5.2}$$

Next, to ensure the diversity of the sample selection, the selection space is divided in each domain with the corresponding allocated budgets. We employ the k-means algorithm to divide the selection space into B_k regions for the k-th domain. We utilize gradients E at the last layer of the model as embeddings. Compared to the original feature space, the gradient space is more discriminative and better represents the sample influence on the current model (Ash, Zhang, et al., 2020). The division process could be written as follows:

$$\{S_{k,1}, \cdots, S_{k,B_k}\} = k\text{-means}(\{E_{k,1}, E_{k,2}, \cdots, E_{k,n_k}\}, B_k) \tag{5.3}$$

where $E_{k,i}$ represents the gradient of the i-th sample in domain k, and $S_{k,j}$ represents the j-th cluster in domain k.

5.3.2 Stage Two: Domain Influence Estimation

Given the established regions for selection, we can evaluate the cross-domain influence of samples in each region, ensuring that the selected samples benefit not only the current domain but also other domains. The sample with the highest evaluation score is selected from each region. This evaluation is based on the characteristic of the ASP-MTL model, where a domain-shared feature extractor $F_s(\cdot)$ and domain-private feature extractors $F_{p_k}(\cdot)$ are combined to form the final feature representation. Since the shared information is captured solely by the shared feature extractor, the cross-domain influence evaluation could base on this component.

The intuition of our method is that if a sample is informative to the shared feature extractor, it will be informative to all domains. In AL, a common approach to evaluating the informativeness of a sample is uncertainty measurement. Motivated by previous works (Ducoffe and Precioso, 2018b; Xu, Evans, et al., 2018; Dai, Liu, et al., 2023) that measure uncertainty by adversarial samples, we introduce a perturbation-based method to estimate the informativeness of each sample on the shared feature extractor. If a sample is more informative to the shared feature extractor, it will be more vulnerable to perturbations, i.e., the perturbed sample will be more likely to be misclassified. Consequently, such examples, which are less learned by the shared extractor, could be more influential to all the domains. We sample perturbations from a Gaussian distribution $\delta \sim \mathcal{N}(0, \sigma^2)$, and add them to the output of the shared feature extractor. The perturbed output probability of an item in the k-th domain is denoted as:

$$Out_k(x, \delta) = C_k((F_s(x) + \delta) \oplus F_{p_k}(x)) \tag{5.4}$$

Here, $\oplus$ denotes the concatenation operation, and C_k represents the classifier for the k-th domain. The distance between the original output and the perturbed output is used to evaluate the cross-domain informativeness of the sample.

$$Score(x) = \mathop{\mathbb{E}}_{\delta}[Distance(Out_i(x), Out_i(x, \delta))] \tag{5.5}$$

The distance evaluation could be implemented by various metrics. In this study, the distance is calculated by the Kullback-Leibler divergence (Csiszár, 1975) between two distributions, which can be written as:

$$Distance(P, Q) = D_{\mathrm{KL}}(P \| Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right) \tag{5.6}$$

The score represents the expected output distance between the original and perturbed output distributions. Empirically, the score is calculated by sampling multiple perturbations.

5.3.3 Overall framework and Pseudocode

Incorporating both stages together, the overall framework of the proposed P2S-MDAL method is shown in Algorithm 2. All the marks in the algorithm are consistent with the Definition 8 in Section 2.3.3.

5.4 Experimental Setup

5.4.1 Research Questions

Given the P2S-MDAL method, the following research questions arise:

1. As the first dedicated AL strategy for MDAL, does P2S-MDAL outperform conventional AL strategies? (Section 5.5.1)

2. P2S-MDAL consists of two stages, to ensure the in-domain diversity and cross-domain informativeness. Whether each stage provides positive effects? (Section 5.5.2)

Algorithm 2 Perturbation-Based Two-Stage Multi-Domain Active Learning (P2S-MDAL)

Require: Initial multi-domain labeled dataset $\mathcal{L}_0$; initial unlabeled multi-domain dataset $\mathcal{U}_0$; multi-domain data pool $\mathcal{P}$; total annotating budget $Budget$; batch annotating budget B; domain number K; number of instances in k-th domain n_k.

Train $F_{\Theta^*(\mathcal{L}_0,\mathcal{P})}$ on $\mathcal{L}_0$ and $\mathcal{P}$ ▷ Initialize the model
repeat(i-th iteration, start from 1)
  ▷ Selection Procedural
  for $k = 1$ to K **do** ▷ For each domain
    ▷ Selecting Regions Establishment
    $B_k = \frac{n_k}{\sum_{l=1}^{K} n_l} \times B$ ▷ Budget allocation for the current domain
    for $j = 1$ to n_k **do** ▷ For each instance
      Calculate its gradient embedding $E_{k,j}$ through $F_{\Theta^*(\mathcal{L}_{i-1},\mathcal{P})}$
    $\{S_{k,1}, \cdots, S_{k,B_k}\} = k\text{-means}(\{E_{k,1}, E_{k,2}, \cdots, E_{k,n_k}\}, B_k)$ ▷ Establish regions through k-means
    ▷ Domain Influence Estimation
    $\mathcal{Q}_{i,k} = \{\}$ ▷ Initialize domain query batch for current domain
    for each cluster $S_{k,h}$ **do** ▷ In each cluster
      for each item x **do**
        $\delta \sim \mathcal{N}(0, \sigma^2)$ ▷ Sample perturbations
        $Score(x) = \mathop{\mathbb{E}}_{\delta}[Distance(Out_i(x), Out_i(x, \delta))]$ ▷ Evaluate instances with distance scores
      Add x with the highest score to $\mathcal{Q}_{i,k}$ ▷ Select from the current cluster
  ▷ AL Status Update Procedure
  $\mathcal{L}_i = \mathcal{L}_{i-1} \cup \mathcal{Q}_i$ and $\mathcal{U}_i = \mathcal{U}_{i-1} \backslash \mathcal{Q}_i$ ▷ Multi-domain dataset update
  Train $F_{\Theta^*(\mathcal{L}_i,\mathcal{P})}$ on $\mathcal{L}_i$ and $\mathcal{P}$ ▷ Multi-domain model update
  $Budget = Budget - B$ ▷ Total budget update
until $Budget$ has been depleted

3. In comparison to other conventional AL strategies, how does the time complexity of P2S-MDAL compare against the others? (Section 5.5.3)

5.4.2 Dataset

Three popular multi-domain textual and image datasets are used in our experiments, namely Amazon (Chen, Xu, et al., 2012), COIL (Nene et al., 1996), and FDUMTL (Liu, Qiu, et al., 2017). All the tasks are classification tasks, and the categories are the same across domains.

- **Amazon** dataset consists of four textual domains: books, dvd, electronics, and kitchen. Each domain contains two categories, with instances encoded to a vector representation of length 5000. There are 2000 samples on each domain. The ratio 6:2:2 is used to split the training/validation/test sets.
- **COIL:** consists of two image domains, each containing twenty categories, with instances encoded to a vector representation of length 1024. There are 720 samples on each domain. The ratio 6:2:2 is used to split the training/validation/test sets.
- **FDUMTL** consists of sixteen textual domains, each containing two categories, and utilizes word2vec embedding for raw texts. In addition to using all the domains, we also conduct evaluations on the first four domains in our experiment, referred to as FDUMTL (4 domains). There are 2000 samples on each domain. The ratio 6:2:2 is used to split the training/validation/test sets.

5.4.3 Model Implementation

ASP-MTL (Liu, Qiu, et al., 2017) is used with the proposed strategy. On datasets COIL and Amazon, single hidden layer MLP with width 64 is used. On dataset FDUMTL, CNN is used as feature extractor with the output dimension 128. The classifier is a fully connected layer for all datasets. The model is trained using an SGD optimizer with batch size 8. Given the

overlap in models and datasets with the comparative study in Chapter 3, we predominantly adopted the hyperparameters from the previous work.

5.4.4 AL Settings

Five conventional single domain AL strategies are selected for the comparison, as we discussed in chapter 3.

- **Random** is the simplest strategy, which randomly selects instances from each domain.
- **Best vs. Second Best (BvSB)** (Joshi et al., 2009), as an uncertainty measurement, selects instances with the greatest difference in prediction probability between the most and second most likely classes.
- **Expected Gradient Length (EGL)** (Settles and Craven, 2008; Zhang, Lease, et al., 2017) is designed for models that can be optimized by gradients. The instances leading to the longest expected gradient length to the last fully connected layer will be selected.
- **Coreset** (Sener and Savarese, 2018) selects instances using a greedy furthest-first search conditioned on the currently labeled examples by using the middle representation.
- **Batch Active learning by Diverse Gradient Embeddings (BADGE)** (Ash, Zhang, et al., 2020) calculates the gradients of the last fully connected layer. A k-means++ initialization is applied to the gradients to ensure the diversity of the selected batch.

5.4.5 Evaluation

During the selection process, we initialized the model for both the Amazon and COIL datasets by annotating 10% of the training set. The total budget allocated for this process was 50% of the training set, with 5% of the training set being selected in each iteration.

For the FDUMTL approach, the total budget was set at 30% of the training set, considering the larger number of domains involved. Regarding the proposed AL strategy, we conducted multiple experiments with perturbations being sampled 20 times, using a standard deviation of 0.01. To ensure robustness and reliability of the results, all experiments were repeated 10 times with different random seeds, and an average performance was presented. The evaluation of the AL strategies was carried out using learning curves. The horizontal axis of the curves represents the number of selected instances, while the vertical axis indicates the model's accuracy on the test set. Additionally, we calculated the area under the learning curve (AULC) as an additional metric for performance assessment. All the AULC results from model-strategy pairs are analyzed using the Mann-Whitney U test (Mann and Whitney, 1947) as we did in Chapter 3.

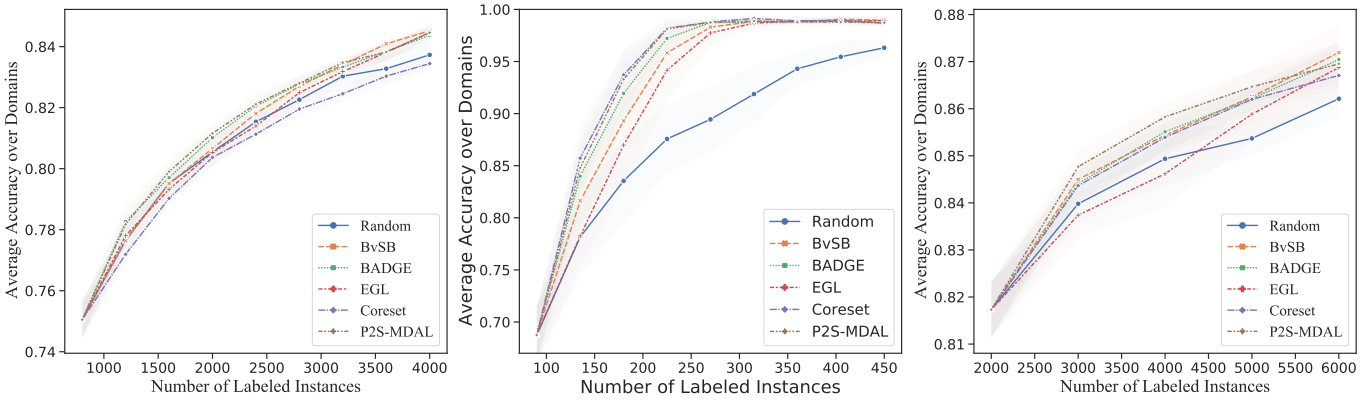
5.5 Results

5.5.1 RQ1: Performance Evaluation

A comparative analysis was conducted between the proposed approach and five active learning (AL) strategies, using three datasets. The results, displayed in Figure 5.2 as learning curves, are further presented in Table 5.1, showing the corresponding ALUC values. The result of statistical significance is shown in appendix C.1.

Table 5.1: Performance in terms of AULC with standard deviation of five conventional AL strategies on three datasets.

Strategy/Datasets	Amazon	COIL	FDUMTL
Random	80.90(0.30)	87.86(2.06)	84.56(0.32)
BvSB	81.20(0.09)	93.20(0.91)	85.16(0.22)
BADGE	81.34(0.15)	94.02(0.82)	85.13(0.21)
EGL	81.04(0.16)	92.18(1.16)	84.64(0.41)
Coreset	80.55(0.14)	**94.64(0.63)**	85.04(0.21)
P2S-MDAL(ours)	**81.40(0.19)**	94.49(0.76)	**85.35(0.18)**



(a) Amazon (b) COIL (c) FDUMTL

Figure 5.2: Performance in terms of learning curves on three datasets, measured by accuracy on the test set with standard deviation.

In the comparison, P2S-MDAL consistently shows top-tier performance across all three datasets. This consistency in performance is especially remarkable for the FDUMTL dataset, which contains sixteen domains and poses increasing complexity in selection difficulty. Despite this challenge, P2S-MDAL still obtains the highest AULC value, further demonstrating its effectiveness. Although the statistical significance of the results is not significant (p-values are larger than 0.05, as shown in Table C.3), there is also no other methods could significantly outperform P2S-MDAL.

Notably, the performance rankings of a single strategy can exhibit significant variation across different datasets. For instance, on the COIL dataset, the proposed P2S-MDAL method and Coreset share the top tier in performance. However, Coreset's performance on the Amazon and FDUMTL datasets is noticeably suboptimal, to the extent of even performing worse than random selection on the Amazon dataset. Thus, the consistent performance of P2S-MDAL across all datasets is a strong indicator of its robustness and reliability.

5.5.2 RQ2: Ablation Study

This section analyzes the effectiveness of both stages of P2S-MDAL, which are designed to ensure the in-domain diversity and cross-domain informativeness. The evaluation is con-

ducted on FDUMTL, a representative and challenging dataset in MDL.

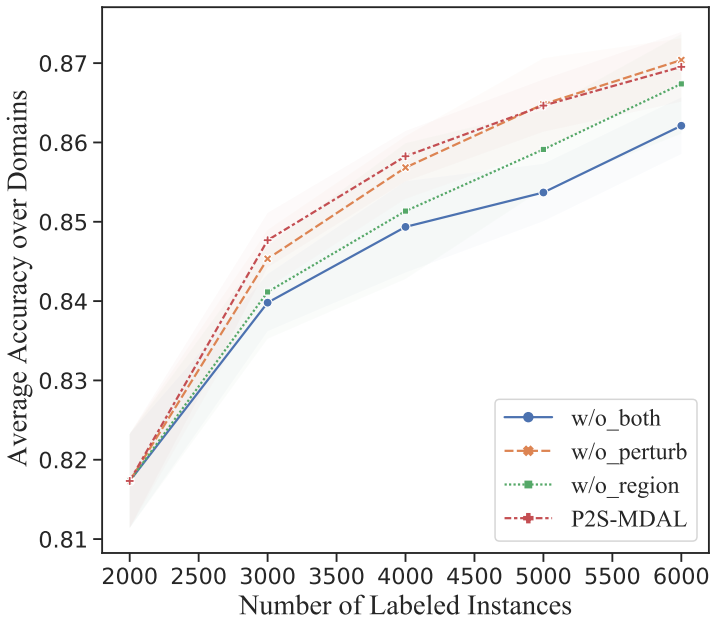


Figure 5.3: Performance in terms of learning curves, an ablation study on FDUMTL. The lighter area represents the standard deviation.

Initially, we discuss the performance of P2S-MDAL without either of its stages. To investigate this, an ablation study is performed, where we systematically remove each stage individually and both stages together, and then compare their respective performances. Specifically, we removed the perturbation module (w/o perturb), the region establishment module (w/o region), and both of them (w/o both), subsequently comparing their performances. In the w/o perturb setup, we replaced the perturbation term using random selection. For the w/o region configuration, we abandoned the k-means pre-clustering and calculated perturbation scores for all instances, subsequently selecting instances with the highest scores. Lastly, the w/o both scenario involved using random selection without either module. The results of this ablation study are illustrated in Figure 5.3, and the statistical significance is shown in Table C.4. In the result, w/o perturb and w/o region are significantly better than w/o both (p-values are larger than 0.05), which indicates the effectiveness of each module. However, although P2S-MDAL obtains largest AULC, it is not significantly better than w/o

perturb.

Table 5.2: Perturbation Analysis on FDUMTL dataset.

Two-Stage Strategy	AULC
2S-Center	85.14 (0.16)
2S-BvSB	85.12 (0.24)
2S-EGL	84.98 (0.21)
P2S-MDAL(ours)	**85.35 (0.18)**

Given there is no significant difference between P2S-MDAL and w/o perturb, we further analyze the effectiveness of the perturbation module (the second stage). As a two-stage method, the first stage of establishing selected regions can technically be paired with various active learning strategies in the second stage. Thus, we utilize the same selecting regions establishment processes (first stage), and substitute different AL strategies (namely Center, BvSB, EGL) at the second stage, denoted as 2S-Center, 2S-BvSB, and 2S-EGL respectively. Specifically, 2S-Center selects the center point in each cluster at the second stage. The performance, measured by the area under the learning curve (AULC), is shown in Table 5.2, and the statistical significance is shown in Table C.5. P2S-MDAL emerges as the top performer in AULC once again, but it is still not significantly better than the others.

5.5.3 RQ3: Time Complexity Analysis

The time complexity is qualitatively analyzed in this section. P2S-MDAL is a two-stage approach, which can be considered a combination of conventional score-based method and distribution-based method. The time complexity for score-based methods (e.g. Uncertainty, EGL) typically falls in $O(n)$, where n represents the quantity of unlabeled instances. P2S-MDAL includes the scoring stage, which means it at least has higher time complexity than score-based methods. Distribution-based methods (e.g. BADGE, Coreset) generally have a high time complexity due to the requisite calculations of pairwise distances. Consequently, these methods become more time-consuming as the number of domains and instances in-

crease. P2S-MDAL also calculates pairwise distances in selecting regions establishment, whereas it is implemented on each domain with a reduced number of items. This leads to significantly shorter processing times when compared to distribution-based methods.

5.6 Chapter Summary & Discussion

This chapter marks the first instance of a dedicated AL strategy designed to address the MDAL problem. The proposed P2S-MDAL method is a two-stage approach, which first establishes selection regions in each domain for sampling diversity, and then selects the most cross-domain influential instances in each region by using perturbations. The efficacy of the proposed method is substantiated by the evaluations on three separate datasets. Although the statistical analysis of the results is not significant, P2S-MDAL still can consistently perform at the top-tier across all datasets compared to other strategies. Furthermore, the ablation study and the perturbation analysis contribute additional information of the effectiveness on both modules in P2S-MDAL. Despite the lack of statistical significance (especially the perturbation part in the ablation study), this work still can be considered as a pioneering work in the field of MDAL.

In the future, this work can be extended in the following directions:

- We anticipate that the budget allocation method could be enhanced by incorporating considerations of data distribution and domain difficulty. With the feedback from either the training or validation set, the budget allocation could be adjusted to better fit the data.

- Moreover, with some adaptations, the proposed perturbation-based cross-domain evaluation could be extended to other MDL models.

- It is worth noting that the perturbation is only applied to the shared feature extractor, which is the only part that is shared across domains. This may be ignoring samples

that are informative to specific domains but not to the others. Thus, the trade-off between the shared and private parts of the model should be further investigated.

- Furthermore, we envision the development of a unified MDAL method, where both models and strategies are designed within single framework. This requires the model training to incorporate the unique characteristics of the AL-selected items. Meanwhile, the AL selection process could benefit from the explicitly designed structure of the model.

Chapter 6

Conclusions

The exploration of learning from multiple domain data represents a practical and significant challenge in the field of machine learning. Multi-domain learning (MDL) leverages knowledge sharing across various domains, mitigating the issue of domain shift, and thereby enhancing model performance. However, MDL also confronts the challenge of high labeling costs, a factor that cannot be overlooked in real-world applications. The labelling burden is accentuated when compared to single-domain instances, due to the need for collaboration among domain experts and complications arising from privacy and legal concerns. In this thesis, we focus on how to apply active learning (AL) to multi-domain data scenarios, specifically termed as multi-domain active learning (MDAL), aiming to alleviate the burden of labeling by selecting the most informative instances for annotation. This inquiry has led to the formulation of three specific research questions (RQs), which are presented in Chapter 1 and addressed in Chapter 3 to Chapter 5. Initially, we propose an MDAL pipeline as an off-the-shelf solution by merging existing MDL techniques with conventional single-domain AL strategies. Subsequently, we introduce a novel plug-and-play contrastive-learning-based method, MDCL, for guaranteeing the performance of MDL models in scenarios with limited labeling. Lastly, we present P2S-MDAL, an innovative MDAL strategy that assesses and annotates multi-domain informative instances. In the following sections, we succinctly sum-

marize the primary content and contributions of the thesis. Additionally, potential avenues for future research will be explored from the broader scope of the entire thesis.

6.1 Thesis Summary

In chapter 1 and 2, we have provided an overview and the RQs of the thesis, reviewed the literature of MDL and AL, and introduced the preliminaries required to understand the thesis. These chapters also present the motivation and significance of the thesis, supporting systematically presents our researches in the following chapters. Moreover, a well-structured open-source active learning knowledge library, named "awesome active learning", is introduced in Section 2.4.2.

In Chapter 3, we first examine the existing literature on MDAL, which was previously introduced in Chapter 2. Our analysis reveals a significant gap in the field, as there is currently no unified solution for MDAL. Moreover, there is a lack of research specifically addressing MDAL, and the existing methods for MDAL cannot be directly applied to new neural-network-based models and tasks. Consequently, no specific MDAL method can be directly utilized to tackle this problem. As a result, we stressed the need for a general solution for MDAL, suggesting that conventional AL methods could be applied with minimal modifications. This leads us to the first research question **(RQ1):** "How do conventional active learning methods perform in multi-domain active learning?" To address this question, we initially introduce a pipeline as an off-the-shelf solution for MDAL, enabling the combination of existing neural-network-based MDL models with conventional AL strategies. Based on this pipeline, we conduct a comprehensive empirical study to evaluate the performance of various well-known AL strategies and popular MDL models. The results demonstrate the efficacy of AL in MDL, as well as the effectiveness of the MDAL pipeline.

Chapter 4 focuses on the model perspective of MDAL, emphasizing the direct impact of model quality on MDAL performance. Traditional MDL methods are primarily designed for supervised learning scenarios, assuming the availability of sufficient annotations. However,

this assumption is impractical in real-world situations. Even in cost-efficient MDAL, the model still needs to be trained with limited annotations initially. Therefore, it is crucial to train MDL models with insufficient annotations. Furthermore, dealing with insufficient annotations in MDL poses more challenges compared to single-domain learning, as both the model complexity and sample space are increased. Although several methods have been proposed to address this issue, they are not directly applicable to MDL settings as they primarily focus on domain adaptation for single target domain performance. This leads us to the second research question **(RQ2):** "From the model training perspective of MDAL, how to train MDL models under insufficient annotations?" To answer this question, we propose a novel model called multi-domain contrastive learning (MDCL), which is a plug-and-play method that can be applied to various models using the renowned share-private architecture. MDCL effectively addresses inter-domain alignment with multi-domain unlabeled instances and intra-domain semantic alignment with insufficient annotations. With this proposed method, we conduct a comprehensive empirical study to evaluate the performance of MDCL on various models and dataset. The results demonstrate the effectiveness of MDCL in MDL settings and the efficacy of the proposed method.

In Chapter 5, we delve into the strategy perspective of MDAL. The previous findings on MDAL were primarily based on conventional single domain AL strategies, as discussed in Chapter 3 and 4. While these strategies offer viable solutions to MDAL problems, they still result in sub-optimal active learning selection due to the inadequate evaluation of domain-shared information for each instance. This deficiency leads us to the third research question **(RQ3):** "From the labeling strategy perspective of MDAL, how to design effective AL strategies tailored explicitly for multi-domain data?" In response to this question, we propose the first ad hoc active learning strategy for MDAL, called perturbation-based two-stage multi-domain active learning (P2S-MDAL), which assesses and annotates multi-domain informative instances. The P2S-MDAL strategy begins with budget allocation and data clustering, followed by the selection of the most influential cross-domain instances within each cluster. Although the statistical analysis of the results is not significant, P2S-MDAL still can con-

sistently perform at the top-tier across all datasets compared to other strategies. This work still can be considered as a pioneering work in the field of MDAL.

6.2 Limitations

Practical considerations in data collection and oracles. In order to streamline the problem and focus on the core aspects of MDAL, our experiments were conducted based on several strong assumptions regarding data collection and oracles. Firstly, we assumed an equal portion of labeled data would be collected from each domain at the beginning of the experiment to facilitate warm starting of AL. However, it is important to note that this assumption may not hold in real-world scenarios, as the initial data collection process can vary significantly across domains due to the variant complexity of the data. Further, warm starting may not be a viable option in domains with extremely limited initial labels (Yuan, Lin, et al., 2020). Furthermore, we assume that the cost and quality of the oracle would remain consistent across domains. However, it is crucial to recognize that this assumption is not valid in real-world scenarios, even when considering single domain learning (Yan, Rosales, et al., 2011). The inherent difficulty of the data associated with multiple domains introduces variations in the cost and quality of oracles across domains. These considerations are not addressed in our experiments for the sake of simplicity.

Quantitative evaluation of the cross-domain information. The objective of an ideal MDAL strategy is to enhance overall performance by selecting the most informative instances across all domains. This selection should be based on a quantitative evaluation of cross-domain information, allowing for comparisons between instances from different domains. However, the MDAL pipeline discussed in Chapter 3 compares instances from different domains, which may lead to incompatible evaluation scores due to the scores being derived from different model components. Furthermore, the P2S-MDAL method introduced in Chapter 5 avoids explicit comparisons between elements from different domains and only

proceeds in-domain evaluation. The absence of cross-domain quantitative evaluation necessitates the inclusion of a budget allocation process to determine the number of instances to be selected from each domain, thereby adding complexity to the strategy. Consequently, the lack of proper quantitative evaluation raises concerns about the sub-optimality of the strategy.

Active learning experimental setup. Throughout this thesis, we conducted numerous experiments in the active learning scenario, employing various empirical settings such as AL-batch size, initial training data size, overall budgets (expressed as percentages), and repeat times (to measure average performance) within the AL experimental setup. We made efforts to maintain consistency in these settings across different experiments, methods, and datasets (Section 3.3, Section 4.4, Section 5.4). Nevertheless, these empirical settings pose challenges when it comes to generalizing the results and facilitating further comparisons. As a consequence, future researchers must carefully design their experimental setups to enable meaningful comparisons. Given varying setups, researchers have to re-implement the methods and re-run all the experiments. This issue has also been mentioned by several previous AL researches (Munjal et al., 2022; Beck et al., 2021).

Setting of hyperparameters. Practical constraints within the active learning process often preclude comprehensive tuning opportunities. This is because there is only one chance to run the iterative selection process as the budget is limited. The common practice is to use the empirical settings from previous researches for convenience, and this practice is also partly adopted in this thesis. While there are numerous potential approaches for hyperparameter tuning, such as optimizing based on performance at a specific iteration of labeling, there can also be many limitations to these methods. It is hard to say which parameter is good for the whole AL process, as the performance of the model can change significantly as the labeling process progresses. Although our methodological choices reflect a balance between experimental rigor and the operational constraints of active learning

frameworks, there should be a more systematic approach to hyperparameter tuning in future AL research.

Setting of Performance Metric. Throughout this thesis, we have used the accuracy as the primary performance metric for the classification task. The reason for this choice is that accuracy is a widely used metric in the field of machine learning, and has been used in many previous researches on the selected datasets. However, accuracy is not always the best metric for evaluating the performance of a model, especially in the case of imbalanced datasets. In the future, it would be beneficial to consider other metrics for multi-domain learning, which could take into account the imbalance from both the domain and class perspectives.

6.3 Future Directions

This thesis presents a comprehensive overview of the MDAL problem and examines it from both the model and strategy perspectives. The insights gained from this analysis provide valuable guidance for future MDAL research, which could be summarized as follows.

- **Task-specific unified MDAL framework.** In this thesis, we focus solely on the classification task, approaching it from both model and strategy perspectives. However, real-world multi-domain tasks can be more complex, such as multi-domain person re-identification (Xiao, Li, et al., 2016), multi-domain recommendation (Zhang, Jin, et al., 2016), and others. In such case, incorporating task-specific models into MDAL is necessary. Designing unified task-specific MDAL frameworks with models is a promising direction.

- **Data stream based MDAL scenario.** This thesis concentrates on the pool-based AL scenario, where unlabeled data is collected in advance and stored in a pool. However, in real-world applications, data collection may occur in a streaming manner, for instance, in financial markers (Barata et al., 2021) and healthcare monitoring (Fu et

al., 2020). Exploring how MDAL can be adapted to the data stream scenario presents an exciting direction.

- **Open environment (domain) AL problems.** We primarily discuss the MDAL problem within a closed environment, where the domains are known in advance. However, in real-world applications, the domains can be open, meaning new domains may emerge in the future. This requires the learned MDL model to be domain-generalizable, which raises new questions related to MDAL.

  - *How can data be collected cost-efficiently to build domain-generalizable models?* This can be formulated as a problem of active learning with domain generalization, which can be a promising direction. Selecting instances that contribute the most to extracting domain-invariant information while potentially disregarding domain-specific information can result in a more generalizable model for new domains at a lower cost.

  - *How can data be cost-efficiently collected from new domains using available models?* Building on the concept of multi-domain generalization, if new domain data is available during the inference stage, it becomes more cost-efficient to annotate a small set of informative data specific to the new domain. This approach can be formulated as a few-shot active learning or a multi-source active domain adaptation scheme, which is also a promising direction.

- **Domain generalizable large language models (LLM) with active learning** Large language models (LLMs) (Brown et al., 2020) have demonstrated remarkable success in natural language processing (NLP) tasks, being able to generate cross-domain human-like contents (Lu et al., 2023), making them a powerful solution for MDL problems. However, the application of these domain-generalizable LLMs to specific real-world tasks, such as giving medical advices (Lee et al., 2023), faces a challenge in efficient model adaptation. Active learning can address this issue in the following

ways:

  - *AL assisted fine-tuning of LLMs.* Fine-tuning is a common approach to utilizing LLMs in downstream tasks (Schröder et al., 2022; Seo et al., 2022). Active learning offers a promising solution by reducing the data requirement for new tasks that have only a few annotated examples.
  - *AL assisted in-context example building.* In-context learning (Brown et al., 2020) empowers LLMs to learn from a small number of examples within a specific context. However, determining how to construct valuable in-context examples remains an open question. Active learning, in combination with the general domain knowledge of LLMs, presents a potential solution to this challenge (Margatina, Schick, et al., 2021).

- **Multi-domain dataset condensation** Dataset condensation (Zhao, Mopuri, et al., n.d.) is a promising approach to build a compact dataset that preserves the performance of the original dataset, which is similar to the goal of active learning. It has been applied to many research fields such as graph processing (Jin et al., 2022), and recommendation system (Wu, Fan, He, et al., 2023). However, the current dataset condensation methods are designed for single-domain settings, and constructing a multi-domain dataset condensation method would be a promising direction.

Appendix A

Supplementary Materials for the Comparative Study

Supplementary Materials for Multi-Domain Active Learning: a Comparative Study in Chapter 3.

A.1 Model Structures

For each dataset, the compared models share the same micro-structure, i.e., the structures of feature extractors, classifiers, and discriminators of different models are set to be the same. The particular structures of modules are presented in Table A.1. For the datasets used in the corresponding paper, the model structures remain. For Amazon, Office-31, Office-Home, and ImageCLEF datasets, shallow neural networks with one hidden layer are used. Deep neural networks are used for Digits and PACS datasets.

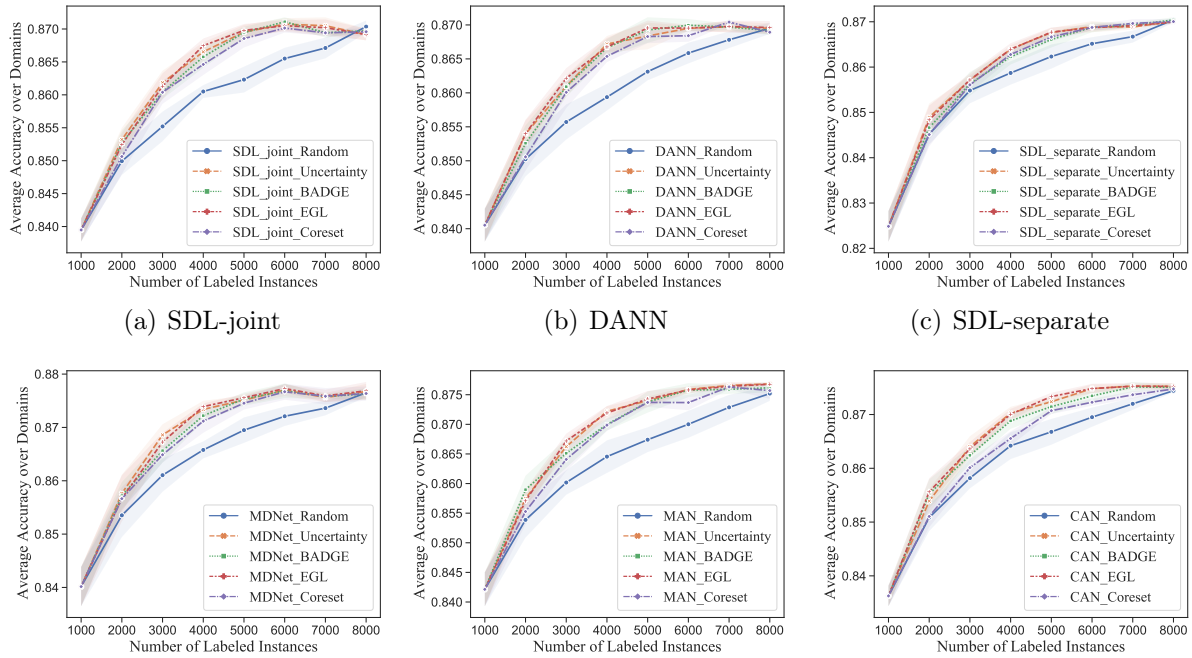
Table A.1: Structures of different modules

Datasets \ Modules	Feature Extractor	Classifier	Discriminator
Amazon	Linear(30000,50) Sigmoid layer	Linear(50,2) Softmax layer	Linear(50,4) Softmax layer
Office-31	Linear(4096,50) Sigmoid layer	Linear(50,31) Softmax layer	Linear(50,3) Softmax layer
Office-Home	Linear(2048,100) Sigmoid layer	Linear(100,65) Softmax layer	Linear(100,4) Softmax layer
ImageCLEF	Linear(1024,50) Sigmoid layer	Linear(50,12) Softmax layer	Linear(50,3) Softmax layer
Digits	Conv2d(3, 32, kernel_size=5) BatchNorm2d(32) nn.MaxPool2d(2) ReLU Conv2d(32, 48, kernel_size=5) BatchNorm2d(48) Dropout2d() MaxPool2d(2) ReLU	Linear(48 * 4 * 4, 100) BatchNorm1d(100) ReLU Dropout Linear(100, 100) BatchNorm1d(100) ReLU Linear(100, 10) Softmax Layer	Linear(48 * 4 * 4, 100) BatchNorm1d(100) ReLU Linear(100, 2) Softmax Layer
PACS	Pre-trained ResNet-18	Linear(512,7) Softmax layer	Linear(512, 1000) ReLU Dropout Linear(1000, 1000) ReLU Linear(1000, 4) Softmax Layer

A.2 Additional Results of Comparisons

A.2.1 Results of Comparisons over Strategies

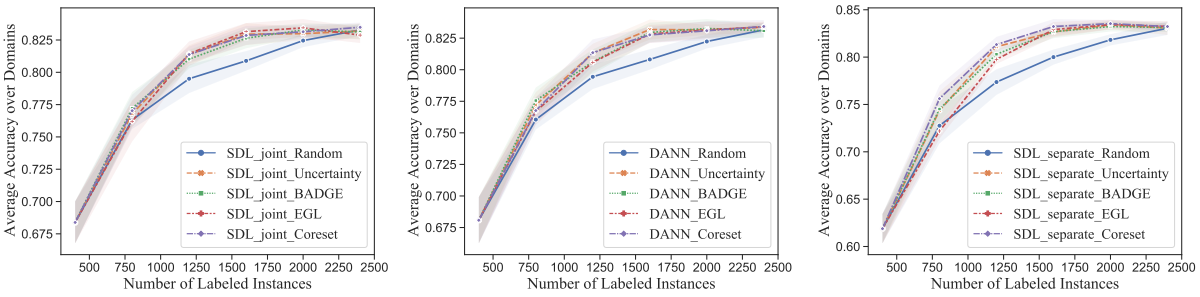
In the main body of the paper, the model-strategy pairs are compared in terms of AULC. The raw learning curves for Amazon, Office-31, Office-Home, ImageCLEF, Digits, and PACS datasets are shown in Fig. A.1 to Fig. A.6, respectively. The learning curves of the well-performed Uncertainty strategy can be found in Fig. A.7. For the imageCLEF dataset, random selection already performs well, and the performance of random selection is presented.



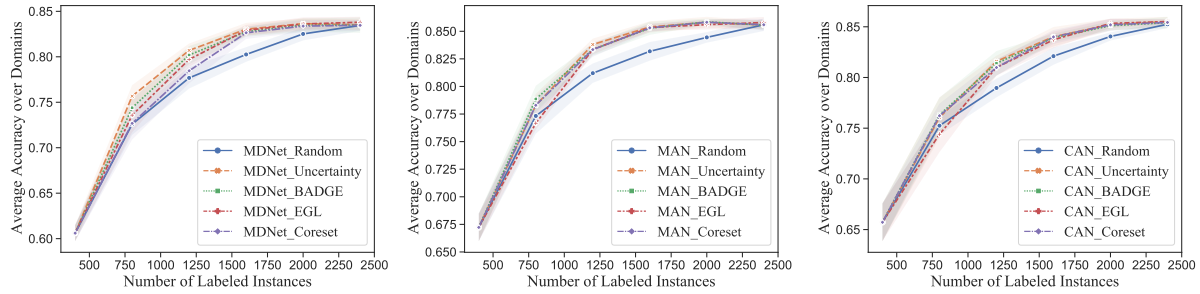
(a) SDL-joint (b) DANN (c) SDL-separate

(d) MDNet (e) MAN (f) CAN

Figure A.1: The performance of model-strategy pairs on the Amazon dataset.

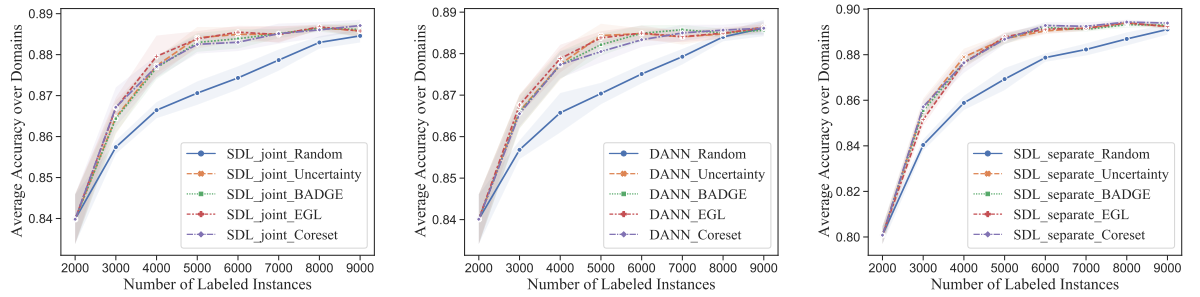


(a) SDL-joint (b) DANN (c) SDL-separate

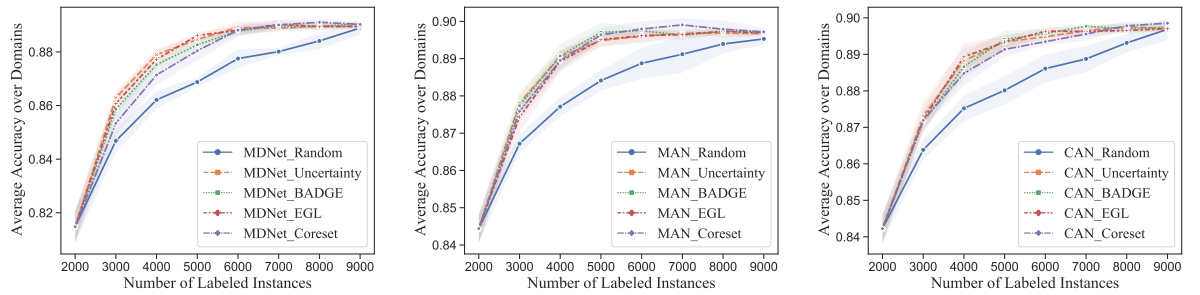


(d) MDNet (e) MAN (f) CAN

Figure A.2: The performance of model-strategy pairs on the Office-31 dataset.

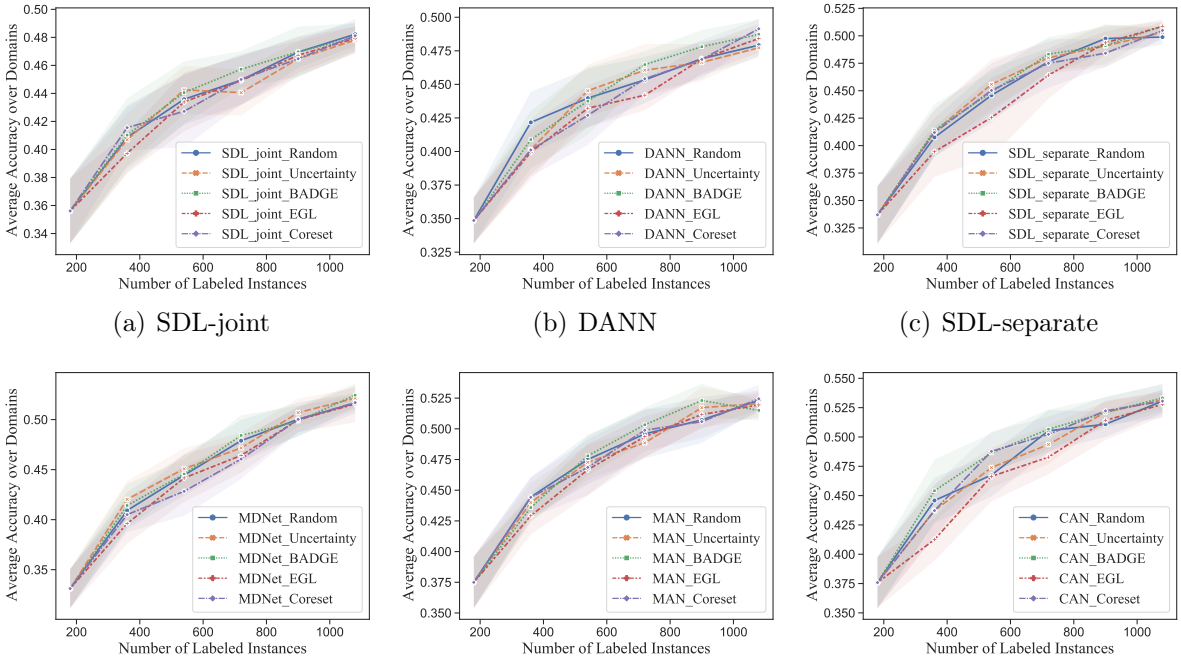


(a) SDL-joint (b) DANN (c) SDL-separate



(d) MDNet (e) MAN (f) CAN

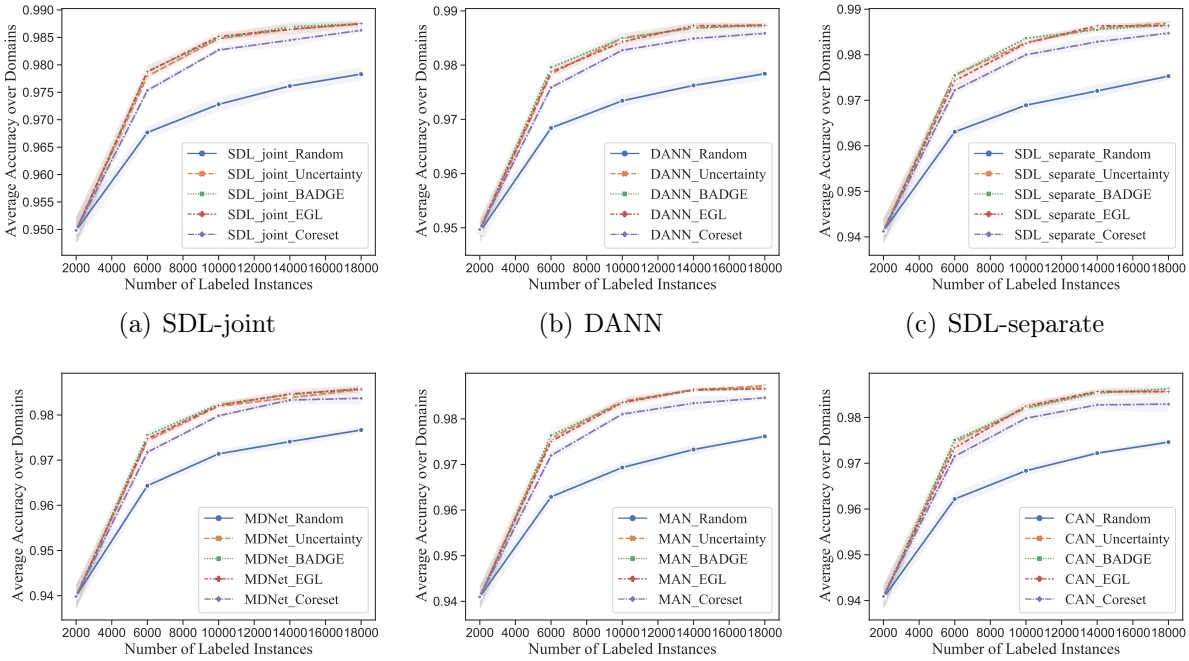
Figure A.3: The performance of model-strategy pairs on the Office-Home dataset.



(a) SDL-joint (b) DANN (c) SDL-separate

(d) MDNet (e) MAN (f) CAN

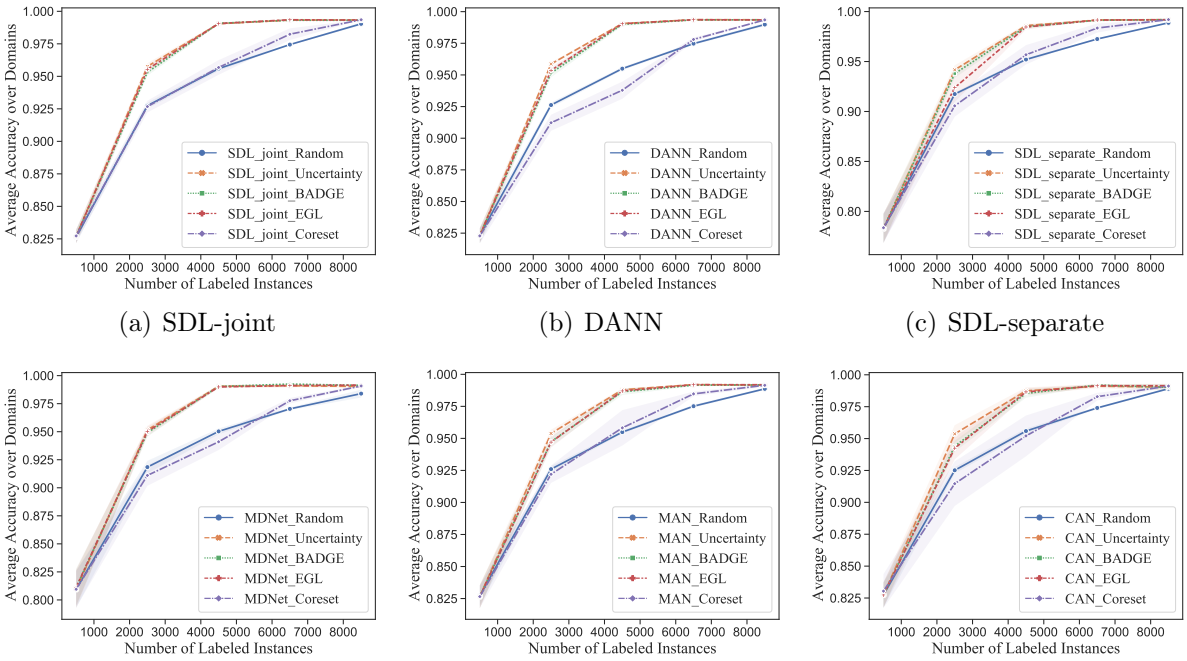
Figure A.4: The performance of model-strategy pairs on the imageCLEF dataset.



(a) SDL-joint (b) DANN (c) SDL-separate

(d) MDNet (e) MAN (f) CAN

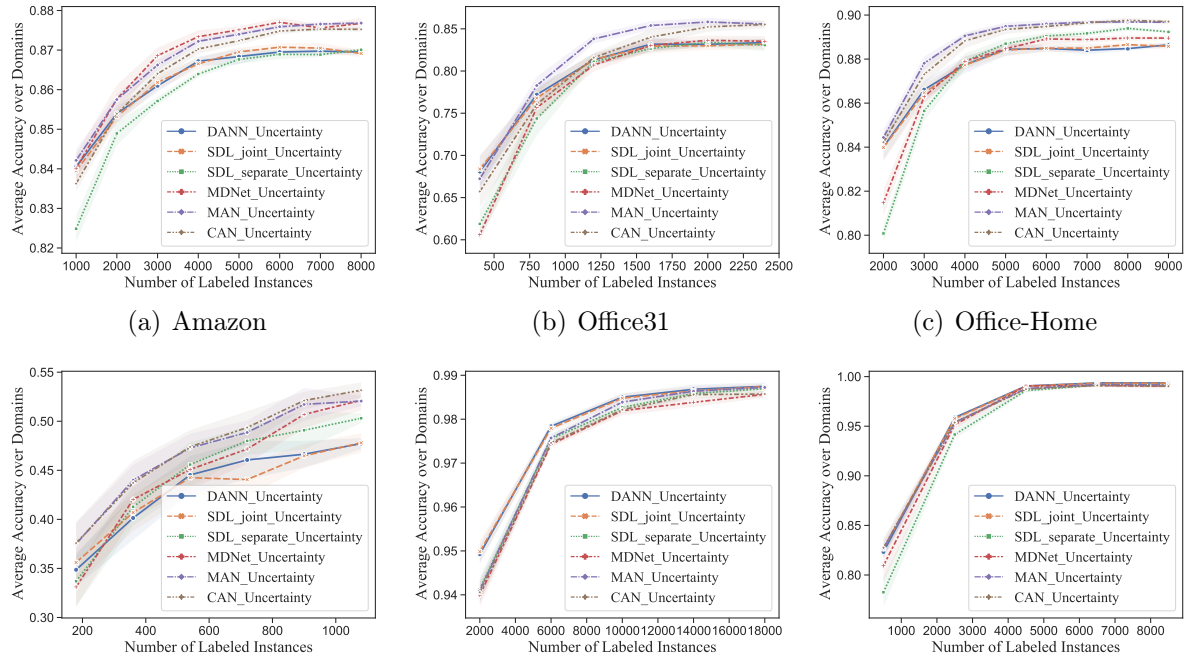
Figure A.5: The performance of model-strategy pairs on the Digits dataset.



(a) SDL-joint (b) DANN (c) SDL-separate

(d) MDNet (e) MAN (f) CAN

Figure A.6: The performance of model-strategy pairs on the PACS dataset.



(a) Amazon (b) Office31 (c) Office-Home

(d) ImageCLEF (e) Digits (f) PACS

Figure A.7: The performance of the Uncertainty strategy on each dataset.

A.2.2 Results of Comparisons on Domains

The domain performances of the Uncertainty strategy are shown in Fig. A.8 to Fig. A.12.

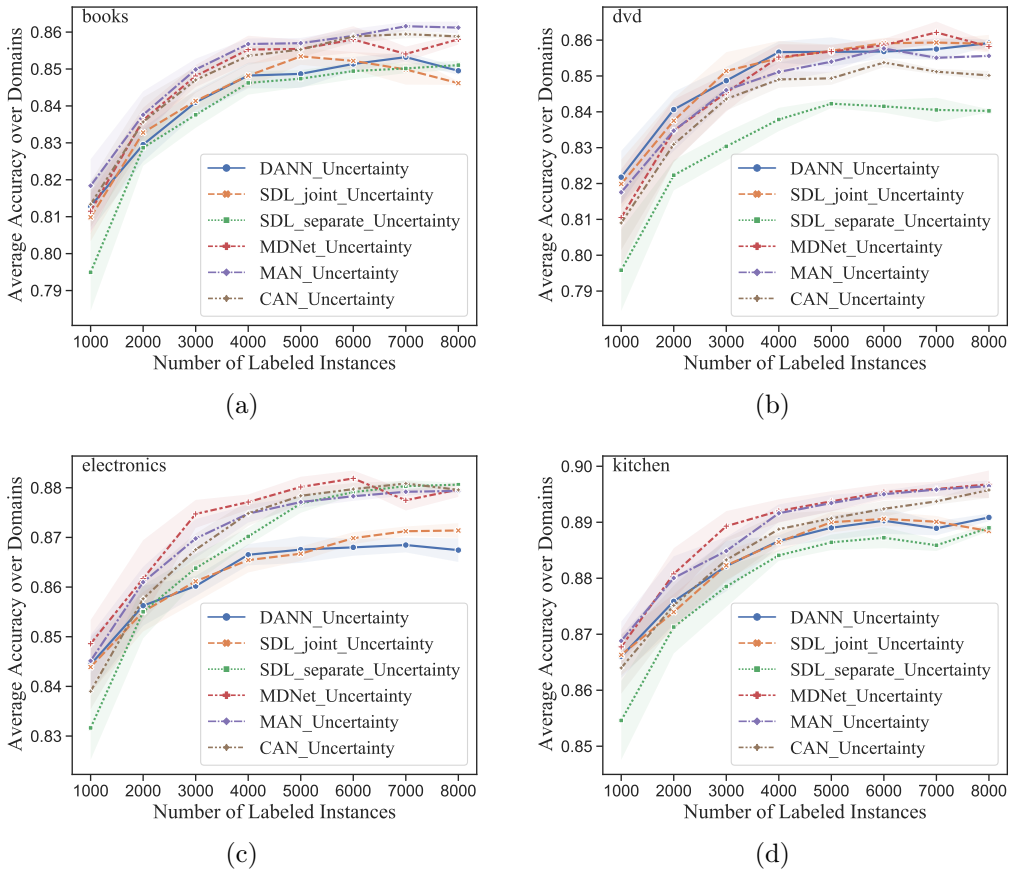


Figure A.8: Learning curves of model-strategy pairs in different domains on Amazon.

A.3 Statistical Analysis

All the AULC results from model-strategy pairs are analyzed using the Mann-Whitney U test (Mann and Whitney, 1947). For each dataset, firstly, the p-values are calculated for each pair of models, which are shown in table A.2 to A.7. Besides, the p-values for each pair of model-strategy pairs are shown in table A.8 to A.13.

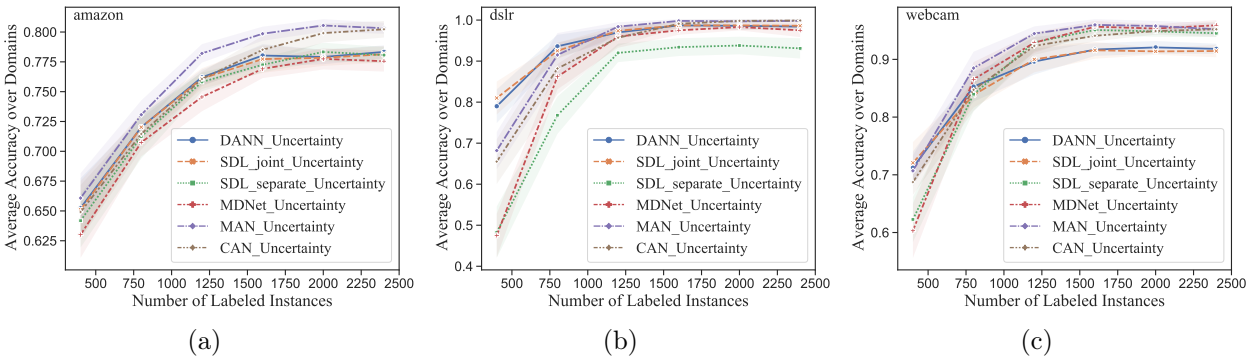


Figure A.9: Learning curves of model-strategy pairs in different domains on Office-31.

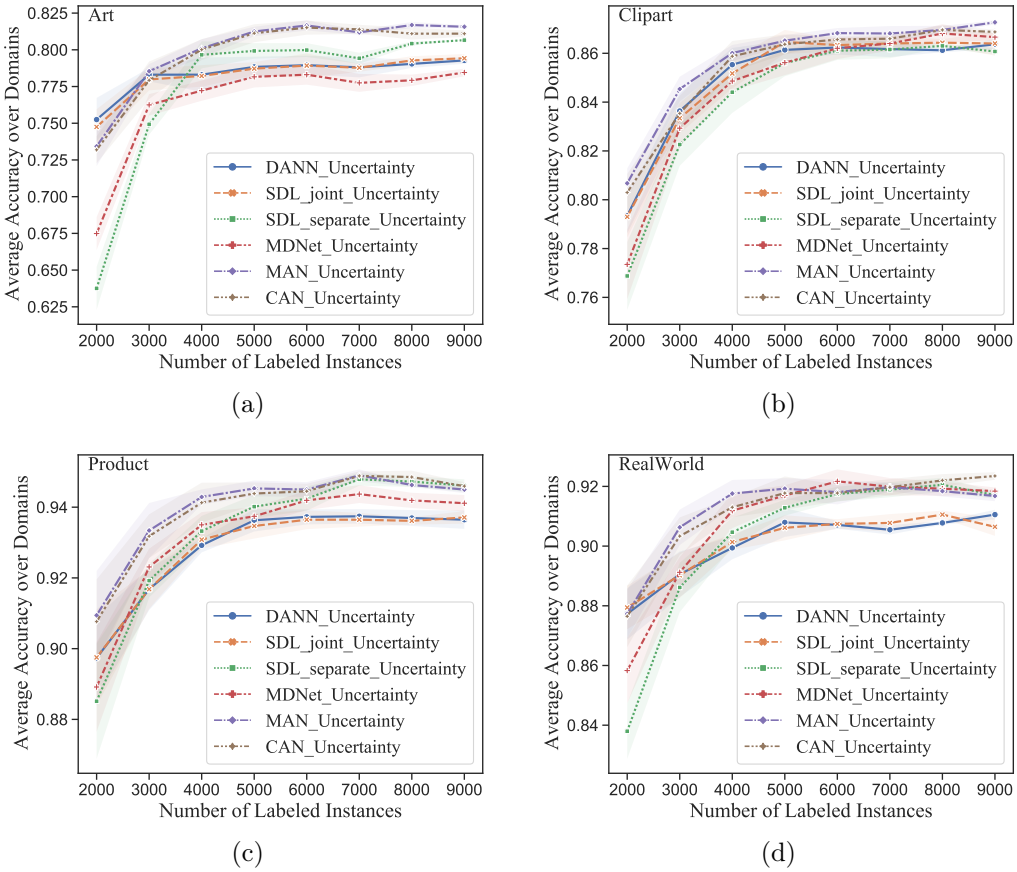


Figure A.10: Learning curves of model-strategy pairs in different domains on Office-Home.

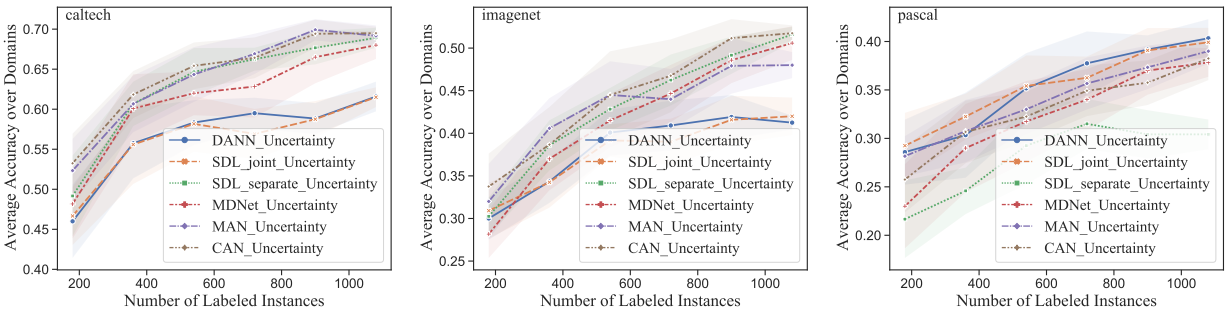


Figure A.11: Learning curves of model-strategy pairs in different domains on ImageCLEF.

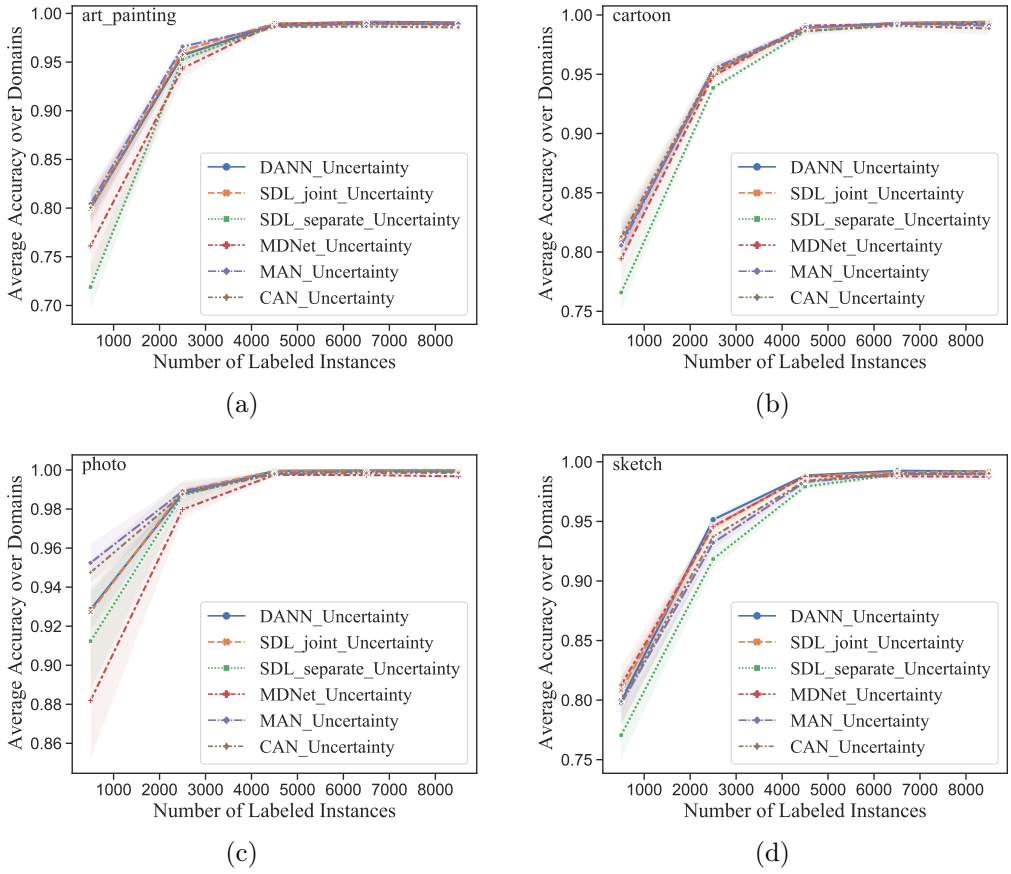


Figure A.12: Learning curves of model-strategy pairs in different domains on PACS.

Table A.2: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **Office-31** dataset.

	DANN	SDL_joint	SDL_separate	MDNet	MAN	CAN
DANN	N/A	0.3256	0.0006	0.0002	0.0002	0.1508
SDL_joint		N/A	0.0003	0.0002	0.0003	0.3071
SDL_separate			N/A	0.6232	0.0002	0.0003
MDNet				N/A	0.0002	0.0002
MAN					N/A	0.0003
CAN						N/A

Table A.3: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **Amazon** dataset.

	DANN	SDL_joint	SDL_separate	MDNet	MAN	CAN
DANN	N/A	0.4727	0.0017	0.0002	0.0002	0.0003
SDL_joint		N/A	0.0036	0.0002	0.0002	0.0003
SDL_separate			N/A	0.0002	0.0002	0.0002
MDNet				N/A	0.273	0.0036
MAN					N/A	0.0211
CAN						N/A

Table A.4: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **Office-Home** dataset.

	DANN	SDL_joint	SDL_separate	MDNet	MAN	CAN
DANN	N/A	0.535	0.0006	0.0041	0.0006	0.0006
SDL_joint		N/A	0.0006	0.0111	0.0006	0.0006
SDL_separate			N/A	0.3829	0.0006	0.0006
MDNet				N/A	0.0006	0.0006
MAN					N/A	0.053
CAN						N/A

Table A.5: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **ImageCLEF** dataset.

	DANN	SDL_joint	SDL_separate	MDNet	MAN	CAN
DANN	N/A	0.4955	0.0376	0.0191	0.0002	0.0002
SDL_joint		N/A	0.0058	0.0046	0.0002	0.0002
SDL_separate			N/A	0.5205	0.0002	0.0004
MDNet				N/A	0.0004	0.0006
MAN					N/A	0.2119
CAN						N/A

Table A.6: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **Digits** dataset.

	DANN	SDL_joint	SDL_separate	MDNet	MAN	CAN
DANN	N/A	0.623	0.0002	0.0002	0.0002	0.0002
SDL_joint		N/A	0.0002	0.0002	0.0002	0.0002
SDL_separate			N/A	0.0008	0.1304	0.1859
MDNet				N/A	0.0028	0.0003
MAN					N/A	0.0211
CAN						N/A

Table A.7: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **PACS** dataset.

	DANN	SDL_joint	SDL_separate	MDNet	MAN	CAN
DANN	N/A	0.1508	0.0079	0.0159	0.6004	0.8413
SDL_joint		N/A	0.0079	0.0079	0.4206	0.5476
SDL_separate			N/A	0.3095	0.0079	0.0079
MDNet				N/A	0.0079	0.0317
MAN					N/A	0.6905
CAN						N/A

Table A.8: The p-values of the Mann-Whitney U test for the **Office-31** dataset. The analysis is conducted on the AULC results of different model-strategy pairs.

Model	Strategy	DANN					SDL_joint					SDL_separate					MDNet					MAN					CAN				
		Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset
DANN	Random	N/A	0.0003	0.0003	0.0009	0.0004	0.3256	0.0004	0.0006	0.0013	0.0003	0.0006	0.9097	0.3847	0.001	0.0493	0.0002	0.1857	0.2729	0.1508	0.0015	0.0002	0.0002	0.0002	0.0002	0.0002	0.1508	0.0002	0.0002	0.0013	0.0003
	Uncertainty		N/A	0.1212	0.0962	0.162	0.0003	0.3073	0.65	0.5966	0.3075	0.0002	0.0003	0.0002	0.0002	0.0025	0.0002	0.0004	0.0002	0.0002	0.0002	0.0757	0.0002	0.0002	0.0002	0.0002	0.0015	0.3075	0.273	0.7623	0.1123
	BADGE			N/A	0.7054	0.65	0.0017	0.545	0.8501	0.6232	0.3447	0.0002	0.0003	0.0002	0.0002	0.0028	0.0002	0.0004	0.0002	0.0002	0.0002	0.0539	0.0002	0.0002	0.0002	0.0002	0.0028	0.0376	0.1508	1	0.082
	EGL				N/A	0.5205	0.0028	0.4053	0.4274	0.3642	0.3447	0.0002	0.0006	0.0006	0.0002	0.0091	0.0002	0.001	0.0002	0.0002	0.0002	0.0065	0.0002	0.0002	0.0002	0.0002	0.0073	0.014	0.0257	0.4274	0.0073
	Coreset					N/A	0.0019	0.7336	1	0.6232	0.9097	0.0002	0.0008	0.0004	0.0002	0.0091	0.0002	0.0008	0.0003	0.0002	0.0002	0.0211	0.0002	0.0002	0.0002	0.0002	0.0065	0.082	0.1986	0.9397	0.0211
SDL (joint)	Random						N/A	0.0011	0.0017	0.0036	0.0008	0.0003	0.7913	0.0757	0.0004	0.1212	0.0002	0.4725	0.0376	0.014	0.0004	0.0003	0.0002	0.0002	0.0002	0.0002	0.3071	0.0004	0.0005	0.0022	0.0003
	Uncertainty							N/A	1	0.7336	1	0.0002	0.0006	0.0004	0.0002	0.0073	0.0002	0.0008	0.0002	0.0002	0.0002	0.0376	0.0002	0.0002	0.0002	0.0002	0.0046	0.0889	0.1984	0.9698	0.0451
	BADGE								N/A	0.85	1	0.0002	0.0003	0.0004	0.0002	0.0058	0.0002	0.0004	0.0002	0.0002	0.0002	0.0312	0.0002	0.0002	0.0002	0.0002	0.0032	0.1041	0.2567	0.9698	0.0539
	EGL									N/A	0.8501	0.0002	0.0022	0.0009	0.0002	0.0091	0.0002	0.0046	0.0004	0.0004	0.0002	0.0539	0.0002	0.0002	0.0002	0.0002	0.0058	0.1735	0.2413	1	0.1405
	Coreset										N/A	0.0002	0.0002	0.0002	0.0002	0.0017	0.0002	0.0002	0.0002	0.0002	0.0002	0.0376	0.0002	0.0002	0.0002	0.0002	0.001	0.1212	0.273	0.8501	0.0539
SDL (separate)	Random											N/A	0.0004	0.0008	0.0022	0.0002	0.6232	0.0002	0.001	0.0022	0.0113	0.0002	0.0002	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002	0.0002
	Uncertainty												N/A	0.4727	0.0091	0.064	0.0002	0.4272	0.2896	0.1123	0.0032	0.0002	0.0002	0.0002	0.0002	0.0002	0.1212	0.0002	0.0002	0.0004	0.0003
	BADGE													N/A	0.0173	0.0081	0.0002	0.0448	0.8501	0.5452	0.0028	0.0002	0.0002	0.0002	0.0002	0.0002	0.0173	0.0002	0.0002	0.0006	0.0002
	EGL														N/A	0.0002	0.0002	0.0006	0.014	0.0342	0.104	0.0002	0.0002	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002	0.0002
	Coreset															N/A	0.0002	0.289	0.0013	0.0009	0.0002	0.0004	0.0002	0.0002	0.0002	0.0002	0.6229	0.0004	0.0004	0.0028	0.0008
MDNet	Random																N/A	0.0002	0.0002	0.0008	0.0101	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Uncertainty																	N/A	0.021	0.0022	0.0004	0.0002	0.0002	0.0002	0.0002	0.0002	0.5204	0.0002	0.0002	0.0008	0.0004
	BADGE																		N/A	0.6775	0.0046	0.0002	0.0002	0.0002	0.0002	0.0002	0.0058	0.0002	0.0002	0.0002	0.0002
	EGL																			N/A	0.0211	0.0002	0.0002	0.0002	0.0002	0.0002	0.0046	0.0002	0.0002	0.0002	0.0002
	Coreset																				N/A	0.0002	0.0002	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002	0.0002
MAN	Random																					N/A	0.0002	0.0002	0.0004	0.0002	0.0003	0.6776	0.65	0.0452	0.7335
	Uncertainty																						N/A	0.8205	0.0233	0.65	0.0002	0.0006	0.0008	0.0002	0.0002
	BADGE																							N/A	0.014	0.5705	0.0002	0.0006	0.0007	0.0002	0.0002
	EGL																								N/A	0.0233	0.0002	0.0022	0.0091	0.0004	0.0002
	Coreset																									N/A	0.0002	0.0008	0.0007	0.0002	0.0002
CAN	Random																										N/A	0.0004	0.0004	0.0017	0.0004
	Uncertainty																											N/A	0.8205	0.1123	0.9698
	BADGE																												N/A	0.162	0.8798
	EGL																													N/A	0.1212
	Coreset																														N/A

Table A.9: The p-values of the Mann-Whitney U test for the **Amazon** dataset. The analysis is conducted on the AULC results of different model-strategy pairs.

Model	Strategy	DANN					SDL_joint					SDL_separate					MDNet					MAN					CAN				
		Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset
DANN	Random	N/A	0.0002	0.0002	0.0002	0.0002	0.4727	0.0002	0.0002	0.0002	0.0002	0.0017	0.0376	1	0.0452	0.9097	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002	0.0002
	Uncertainty		N/A	0.3642	0.1986	0.0004	0.0002	0.1735	0.4727	0.3642	0.0058	0.0002	0.0002	0.0002	0.0002	0.0002	0.0113	0.0002	0.0002	0.0002	0.0002	0.1859	0.0002	0.0002	0.0002	0.0002	0.0173	0.0002	0.0002	0.0002	0.0452
	BADGE			N/A	0.082	0.0091	0.0002	0.1041	0.9698	0.162	0.0376	0.0002	0.0002	0.0002	0.0002	0.0002	0.014	0.0002	0.0002	0.0002	0.0002	0.2123	0.0002	0.0002	0.0002	0.0002	0.0452	0.0002	0.0003	0.0002	0.0312
	EGL				N/A	0.0002	0.0002	0.7054	0.064	0.8501	0.0017	0.0002	0.0002	0.0002	0.0002	0.0002	0.0257	0.0002	0.0002	0.0002	0.0002	0.3847	0.0002	0.0002	0.0002	0.0002	0.0058	0.0002	0.0003	0.0002	0.3075
	Coreset					N/A	0.0002	0.0003	0.0058	0.0022	0.9698	0.0002	0.0002	0.0002	0.0002	0.0002	0.0017	0.0002	0.0002	0.0002	0.0002	0.014	0.0002	0.0002	0.0002	0.0002	0.5966	0.0002	0.0002	0.0002	0.0003
SDL_joint	Random						N/A	0.0002	0.0002	0.0002	0.0002	0.0036	0.0173	0.5708	0.0312	0.7337	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002	0.0002
	Uncertainty							N/A	0.0757	0.7623	0.0036	0.0002	0.0002	0.0002	0.0002	0.0002	0.0312	0.0002	0.0002	0.0002	0.0002	0.5205	0.0002	0.0002	0.0002	0.0002	0.0051	0.0002	0.0003	0.0002	0.273
	BADGE								N/A	0.2413	0.0211	0.0002	0.0002	0.0002	0.0002	0.0002	0.0058	0.0002	0.0002	0.0002	0.0002	0.089	0.0002	0.0002	0.0002	0.0002	0.0257	0.0002	0.0002	0.0002	0.0173
	EGL									N/A	0.0058	0.0002	0.0002	0.0002	0.0002	0.0002	0.0257	0.0002	0.0002	0.0002	0.0002	0.4274	0.0002	0.0002	0.0002	0.0002	0.0073	0.0002	0.0003	0.0002	0.2413
	Coreset										N/A	0.0002	0.0002	0.0002	0.0002	0.0002	0.0036	0.0002	0.0002	0.0002	0.0002	0.0312	0.0002	0.0002	0.0002	0.0002	0.65	0.0002	0.0002	0.0002	0.001
SDL_separate	Random											N/A	0.0004	0.0013	0.0004	0.0017	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Uncertainty												N/A	0.0452	0.8501	0.0211	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0013	0.0002	0.0002	0.0002	0.0002
	BADGE													N/A	0.0757	0.7913	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0006	0.0002	0.0002	0.0002	0.0002
	EGL														N/A	0.0211	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.001	0.0002	0.0002	0.0002	0.0002
	Coreset															N/A	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0004	0.0002	0.0002	0.0002	0.0002
MDNet	Random																N/A	0.0002	0.0002	0.0002	0.0002	0.273	0.0002	0.0002	0.0002	0.0002	0.0036	0.0073	0.0312	0.0006	0.273
	Uncertainty																	N/A	0.1405	1	0.0312	0.0002	0.1041	0.014	0.2413	0.0003	0.0002	0.0002	0.0002	0.0002	0.0002
	BADGE																		N/A	0.273	0.1986	0.0002	0.9097	0.3447	0.8501	0.0017	0.0002	0.0002	0.0002	0.0003	0.0002
	EGL																			N/A	0.0452	0.0002	0.3447	0.082	0.3847	0.0004	0.0002	0.0002	0.0002	0.0002	0.0002
	Coreset																				N/A	0.0002	0.064	0.6232	0.089	0.0173	0.0002	0.0013	0.0004	0.0028	0.0002
MAN	Random																					N/A	0.0002	0.0002	0.0002	0.0002	0.0211	0.0006	0.0013	0.0003	0.5708
	Uncertainty																						N/A	0.1212	0.7913	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	BADGE																							N/A	0.2123	0.0081	0.0002	0.0006	0.0006	0.0017	0.0002
	EGL																								N/A	0.0004	0.0002	0.0002	0.0002	0.0002	0.0002
	Coreset																									N/A	0.0002	0.0312	0.0006	0.0757	0.0002
CAN	Random																										N/A	0.0002	0.0002	0.0002	0.0058
	Uncertainty																											N/A	0.1859	0.3847	0.0002
	BADGE																												N/A	0.0257	0.0002
	EGL																													N/A	0.0002
	Coreset																														N/A

Table A.10: The p-values of the Mann-Whitney U test for the **Office-Home** dataset. The analysis is conducted on the AULC results of different model-strategy pairs.

Model	Strategy	DANN					SDL_joint					SDL_separate					MDNet					MAN					CAN				
		Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset
DANN	Random	N/A	0.0006	0.0006	0.0006	0.0006	0.535	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0021	0.0021	0.0006	0.0041	0.0006	0.0006	0.0006	0.0012	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
	Uncertainty		N/A	0.9015	0.535	0.4557	0.0006	0.7104	0.8048	0.3176	0.7104	0.0006	1	0.8982	0.4413	0.7104	0.0006	0.8048	0.1649	0.8048	0.0041	0.0012	0.0006	0.0006	0.0006	0.0006	0.2086	0.0006	0.0006	0.0006	0.0006
	BADGE			N/A	0.3176	0.535	0.0006	0.4557	1	0.2086	0.3829	0.0006	0.62	1	0.4413	0.3176	0.0006	0.3829	0.2593	0.9015	0.0041	0.0012	0.0006	0.0006	0.0006	0.0006	0.0973	0.0006	0.0006	0.0006	0.0006
	EGL				N/A	0.0474	0.0006	0.4817	0.1649	0.4557	0.535	0.0006	0.62	0.0733	0.0542	1	0.0006	0.62	0.0175	0.3829	0.0012	0.0006	0.0006	0.0006	0.0006	0.0006	0.0973	0.0006	0.0006	0.0006	0.0006
	Coreset					N/A	0.0006	0.1649	0.535	0.1282	0.1649	0.0006	0.2086	0.5224	0.7002	0.0973	0.0006	0.053	0.3176	0.3176	0.007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0262	0.0006	0.0006	0.0006	0.0006
SDL_joint	Random						N/A	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0021	0.0021	0.0006	0.0111	0.0006	0.0006	0.0006	0.0012	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
	Uncertainty							N/A	0.2086	0.535	1	0.0006	0.8048	0.0297	0.0291	0.8478	0.0006	0.949	0.0175	0.535	0.0041	0.0006	0.0006	0.0006	0.0006	0.0006	0.053	0.0006	0.0006	0.0006	0.0006
	BADGE								N/A	0.2086	0.1649	0.0006	0.4557	1	0.2481	0.1282	0.0006	0.1413	0.2086	0.6544	0.0041	0.0006	0.0006	0.0006	0.0006	0.0006	0.0262	0.0006	0.0006	0.0006	0.0006
	EGL									N/A	0.3176	0.0006	0.2593	0.2008	0.1235	0.4557	0.0006	0.3829	0.0379	0.1649	0.0023	0.0023	0.0006	0.0006	0.0006	0.0006	0.7491	0.0006	0.0006	0.0006	0.0006
	Coreset										N/A	0.0006	0.949	0.0472	0.0291	0.535	0.0006	0.9015	0.0379	0.5649	0.0175	0.0006	0.0006	0.0006	0.0006	0.0006	0.0474	0.0006	0.0006	0.0006	0.0006
SDL_separate	Random											N/A	0.0006	0.0021	0.0021	0.0006	0.3829	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
	Uncertainty												N/A	0.2764	0.1235	0.3829	0.0006	0.8048	0.053	0.8478	0.0023	0.0006	0.0006	0.0006	0.0006	0.0006	0.053	0.0006	0.0006	0.0006	0.0006
	BADGE													N/A	0.1231	0.0297	0.0021	0.0297	0.3062	0.7489	0.0087	0.0021	0.0021	0.0021	0.0021	0.0021	0.0297	0.0021	0.0021	0.0021	0.0021
	EGL														N/A	0.0209	0.0021	0.0209	0.8463	0.2481	0.0209	0.0021	0.0021	0.0021	0.0021	0.0021	0.0209	0.0021	0.0021	0.0021	0.0021
	Coreset															N/A	0.0006	0.6544	0.007	0.4557	0.0012	0.0006	0.0006	0.0006	0.0006	0.0006	0.2243	0.0006	0.0006	0.0006	0.0006
MDNet	Random																N/A	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
	Uncertainty																	N/A	0.0126	0.4817	0.0012	0.0006	0.0006	0.0006	0.0006	0.0006	0.053	0.0006	0.0006	0.0006	0.0006
	BADGE																		N/A	0.1282	0.0973	0.0006	0.0006	0.0006	0.0006	0.0006	0.007	0.0006	0.0006	0.0006	0.0006
	EGL																			N/A	0.0041	0.0006	0.0006	0.0006	0.0006	0.0006	0.0728	0.0006	0.0006	0.0006	0.0006
	Coreset																				N/A	0.0006	0.0006	0.0006	0.0006	0.0006	0.0012	0.0006	0.0006	0.0006	0.0006
MAN	Random																					N/A	0.0006	0.0006	0.0006	0.0006	0.053	0.0006	0.0006	0.0006	0.0023
	Uncertainty																						N/A	0.3829	0.2593	0.4817	0.0006	0.0175	0.007	0.0006	0.0006
	BADGE																							N/A	0.0379	0.9015	0.0006	0.0111	0.0041	0.0006	0.0006
	EGL																								N/A	0.0728	0.0006	0.1649	0.053	0.0973	0.0012
	Coreset																									N/A	0.0006	0.007	0.0023	0.0006	0.0006
CAN	Random																										N/A	0.0006	0.0006	0.0006	0.0006
	Uncertainty																											N/A	0.8478	0.535	0.0973
	BADGE																												N/A	0.5649	0.1413
	EGL																													N/A	0.0111
	Coreset																														N/A

Table A.11: The p-values of the Mann-Whitney U test for the **ImageCLEF** dataset. The analysis is conducted on the AULC results of different model-strategy pairs.

Model	Strategy	DANN					SDL_joint					SDL_separate					MDNet					MAN					CAN				
		Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset
DANN	Random	N/A	0.6232	0.9698	0.0376	0.1857	0.4955	0.1209	0.9698	0.082	0.4727	0.0376	0.0211	0.0232	0.85	0.064	0.0191	0.001	0.0058	0.1733	0.3642	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0004	0.0002
	Uncertainty		N/A	0.3075	0.2121	0.2565	0.7052	0.3256	0.7622	0.273	1	0.0172	0.0046	0.0073	0.7335	0.0173	0.0046	0.0004	0.001	0.0451	0.1212	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	BADGE			N/A	0.0045	0.0961	0.3445	0.089	0.4725	0.162	0.3256	0.1301	0.0257	0.0211	0.6229	0.1041	0.0233	0.0058	0.0155	0.1733	0.4495	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0008	0.0002
	EGL				N/A	0.6494	0.1617	0.3254	0.1397	0.3073	0.1857	0.0022	0.0004	0.0004	0.1615	0.001	0.0028	0.0002	0.0003	0.0013	0.0028	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Coreset					N/A	0.5449	0.9097	0.2894	0.8501	0.4725	0.0036	0.0028	0.0046	0.5201	0.0081	0.0028	0.0003	0.0006	0.0139	0.0342	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
SDL_joint	Random						N/A	0.3254	0.4723	0.3073	0.9097	0.0058	0.004	0.0058	0.7334	0.0211	0.0046	0.0003	0.0004	0.0172	0.0694	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Uncertainty							N/A	0.344	0.8205	0.4727	0.0058	0.001	0.0028	0.427	0.0126	0.0028	0.0002	0.0002	0.001	0.0091	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	BADGE								N/A	0.2896	0.7334	0.0376	0.0211	0.0154	0.8499	0.0451	0.0376	0.0032	0.0058	0.2119	0.2729	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0006	0.0002
	EGL									N/A	0.4727	0.0073	0.001	0.004	0.2119	0.014	0.0017	0.0002	0.0002	0.0007	0.0113	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Coreset										N/A	0.0113	0.0058	0.0091	0.6229	0.0211	0.0036	0.0006	0.0017	0.0538	0.162	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
SDL_separate	Random											N/A	0.6232	0.5204	0.1402	0.9698	0.5205	0.241	0.3447	0.1733	0.162	0.0002	0.0003	0.0002	0.0003	0.0002	0.0004	0.0002	0.0002	0.0081	0.0002
	Uncertainty												N/A	0.9698	0.0491	0.5452	0.9397	0.5708	0.7623	0.0756	0.0587	0.0003	0.0008	0.0002	0.0008	0.0004	0.0006	0.0007	0.0002	0.014	0.0003
	BADGE													N/A	0.1037	0.2408	0.8797	0.8797	0.623	0.0693	0.0637	0.0002	0.0006	0.0002	0.0006	0.0002	0.001	0.0003	0.0002	0.019	0.0002
	EGL														N/A	0.1855	0.0537	0.021	0.0172	0.3442	0.4051	0.0003	0.0004	0.0002	0.0003	0.0002	0.0003	0.0004	0.0002	0.0017	0.0002
	Coreset															N/A	0.6232	0.2413	0.273	0.3073	0.273	0.0002	0.0002	0.0002	0.0002	0.0002	0.0004	0.0002	0.0002	0.0058	0.0002
MDNet	Random																N/A	0.4055	0.9397	0.0639	0.082	0.0004	0.0006	0.0002	0.0006	0.0003	0.0006	0.0008	0.0002	0.0173	0.0003
	Uncertainty																	N/A	0.9097	0.0065	0.0073	0.0004	0.0004	0.0002	0.001	0.0006	0.0013	0.0007	0.0002	0.0257	0.0003
	BADGE																		N/A	0.0211	0.0173	0.0003	0.001	0.0002	0.001	0.0003	0.0008	0.0013	0.0002	0.0233	0.0004
	EGL																			N/A	0.7049	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0007	0.0002
	Coreset																				N/A	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0007	0.0002
MAN	Random																					N/A	0.8501	0.1859	0.3634	0.9397	0.2119	0.8501	0.082	0.1735	0.2413
	Uncertainty																						N/A	0.3642	0.344	0.8798	0.4723	0.5966	0.0587	0.1986	0.1859
	BADGE																							N/A	0.034	0.2413	0.9698	0.9097	0.1405	0.0126	0.6232
	EGL																								N/A	0.3827	0.037	0.1852	0.0171	0.7333	0.0373
	Coreset																									N/A	0.2119	0.7623	0.0539	0.1859	0.1735
CAN	Random																										N/A	0.85	0.1402	0.0172	0.6229
	Uncertainty																											N/A	0.1508	0.0887	0.3256
	BADGE																												N/A	0.0073	0.4727
	EGL																													N/A	0.0126
	Coreset																														N/A

Table A.12: The p-values of the Mann-Whitney U test for the **Digits** dataset. The analysis is conducted on the AULC results of different model-strategy pairs.

Model	Strategy	DANN					SDL_joint					SDL_separate					MDNet					MAN					CAN				
		Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset
DANN	Random	N/A	0.0002	0.0002	0.0002	0.0002	0.623	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Uncertainty		N/A	0.2567	0.5205	0.0002	0.0002	0.5452	0.5708	0.7913	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	BADGE			N/A	0.3847	0.0002	0.0002	0.0452	0.5708	0.6776	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	EGL				N/A	0.0002	0.0002	0.3847	0.7337	1	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Coreset					N/A	0.0002	0.0002	0.0002	0.0002	0.4495	0.0002	0.0257	0.0046	0.0017	0.0002	0.0002	0.0002	0.0004	0.0002	0.0002	0.0002	0.162	0.2567	0.0091	0.0002	0.0002	0.0003	0.0006	0.0003	0.0002
SDL_joint	Random						N/A	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Uncertainty							N/A	0.1859	0.3447	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	BADGE								N/A	1	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	EGL									N/A	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
	Coreset										N/A	0.0002	0.0342	0.0257	0.0028	0.0002	0.0002	0.0004	0.0006	0.0002	0.0002	0.0002	0.273	0.4274	0.0256	0.0002	0.0002	0.001	0.0017	0.0004	0.0002
SDL_separate	Random											N/A	0.0002	0.0002	0.0002	0.0002	0.0008	0.0002	0.0002	0.0002	0.0002	0.1304	0.0002	0.0002	0.0002	0.0002	0.1859	0.0002	0.0002	0.0002	0.0002
	Uncertainty												N/A	0.6776	0.1508	0.0002	0.0002	0.0036	0.0211	0.001	0.0002	0.0002	0.1041	0.1859	1	0.0002	0.0002	0.1405	0.1986	0.0091	0.0002
	BADGE													N/A	0.1859	0.0002	0.0002	0.001	0.0058	0.0002	0.0002	0.0002	0.0376	0.0539	0.7054	0.0002	0.0002	0.0058	0.0211	0.0028	0.0002
	EGL														N/A	0.0002	0.0002	0.0113	0.2123	0.0312	0.0002	0.0002	0.0046	0.0073	0.1405	0.0002	0.0002	0.3447	0.6776	0.0537	0.0002
	Coreset															N/A	0.0002	0.0028	0.0003	0.0003	0.2727	0.0002	0.0002	0.0002	0.0002	0.1859	0.0002	0.0004	0.0003	0.0002	0.1986
MDNet	Random																N/A	0.0002	0.0002	0.0002	0.0002	0.0028	0.0002	0.0002	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002	0.0002
	Uncertainty																	N/A	0.1041	0.2413	0.001	0.0002	0.0003	0.0003	0.0008	0.064	0.0002	0.1041	0.0539	0.2567	0.0002
	BADGE																		N/A	0.4274	0.0002	0.0002	0.0008	0.001	0.0017	0.0006	0.0002	0.7337	0.4274	0.4274	0.0002
	EGL																			N/A	0.0002	0.0002	0.0002	0.0002	0.0008	0.0002	0.0002	0.1405	0.064	0.7913	0.0002
	Coreset																				N/A	0.0002	0.0002	0.0002	0.0002	0.021	0.0002	0.0002	0.0002	0.0002	0.7912
MAN	Random																					N/A	0.0002	0.0002	0.0002	0.0002	0.0211	0.0002	0.0002	0.0002	0.0002
	Uncertainty																						N/A	0.6776	0.1212	0.0002	0.0002	0.0004	0.0022	0.0002	0.0002
	BADGE																							N/A	0.1735	0.0002	0.0002	0.0006	0.0022	0.0003	0.0002
	EGL																								N/A	0.0002	0.0002	0.0058	0.0211	0.0013	0.0002
	Coreset																									N/A	0.0002	0.0002	0.0002	0.0002	0.0173
CAN	Random																										N/A	0.0002	0.0002	0.0002	0.0002
	Uncertainty																											N/A	0.7337	0.6232	0.0002
	BADGE																												N/A	0.273	0.0002
	EGL																													N/A	0.0002
	Coreset																														N/A

Table A.13: The p-values of the Mann-Whitney U test for the **PACS** dataset. The analysis is conducted on the AULC results of different model-strategy pairs.

Model	Strategy	DANN					SDL_joint					SDL_separate					MDNet					MAN					CAN				
		Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset	Random	Uncertainty	BADGE	EGL	Coreset
DANN	Random	N/A	0.0079	0.0079	0.0079	0.0079	0.1508	0.0079	0.0079	0.0079	0.0952	0.0079	0.0079	0.0079	0.1508	0.0317	0.0159	0.0079	0.0079	0.0079	0.0159	0.6004	0.0079	0.0079	0.0079	0.1508	0.8413	0.0079	0.0079	0.0079	1
	Uncertainty		N/A	0.0952	0.0952	0.0079	0.0079	0.6905	0.1508	0.5476	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0159	0.0159	0.0159	0.0079	0.0079	0.0952	0.0159	0.0079	0.0079	0.0079	0.4206	0.0079	0.0317	0.0079
	BADGE			N/A	0.4206	0.0079	0.0079	0.0556	0.4206	0.2222	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.2222	0.1508	0.2222	0.0079	0.0079	1	0.1508	0.0556	0.0079	0.0079	0.6905	0.0952	0.4206	0.0079
	EGL				N/A	0.0079	0.0079	0.0952	0.4206	0.1508	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0952	0.0952	0.0952	0.0079	0.0079	0.5476	0.0317	0.0159	0.0079	0.0079	1	0.0317	0.0952	0.0079
	Coreset					N/A	0.0079	0.0079	0.0079	0.0079	0.0079	0.5476	0.0079	0.0079	0.0159	0.8413	1	0.0079	0.0079	0.0079	0.5476	0.0079	0.0079	0.0079	0.0079	0.0159	0.0159	0.0079	0.0079	0.0079	0.1508
SDL_joint	Random						N/A	0.0079	0.0079	0.0079	0.4206	0.0079	0.0079	0.0079	0.1508	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.4206	0.0079	0.0079	0.0079	0.1508	0.5476	0.0079	0.0079	0.0079	0.8413
	Uncertainty							N/A	0.1161	0.4206	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0159	0.0159	0.0159	0.0079	0.0079	0.0556	0.0159	0.0079	0.0079	0.0079	0.2222	0.0079	0.0317	0.0079
	BADGE								N/A	0.7533	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0317	0.0317	0.0317	0.0079	0.0079	0.4206	0.0317	0.0159	0.0079	0.0079	1	0.0317	0.0952	0.0079
	EGL									N/A	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0317	0.0159	0.0278	0.0079	0.0079	0.1508	0.0159	0.0079	0.0079	0.0079	0.5476	0.0317	0.0317	0.0079
	Coreset										N/A	0.0079	0.0159	0.0317	0.4206	0.0159	0.0079	0.0079	0.0079	0.0079	0.016	0.2222	0.0079	0.0079	0.0079	1	0.1508	0.0079	0.0079	0.0079	0.5476
SDL_separate	Random											N/A	0.0079	0.0079	0.0079	0.6905	0.3095	0.0079	0.0079	0.0079	0.8413	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079	0.1508
	Uncertainty												N/A	0.4206	0.2222	0.0079	0.0079	0.0159	0.0159	0.0317	0.0079	0.0079	0.0079	0.0079	0.0079	0.0159	0.0079	0.0159	0.0317	0.0159	0.0317
	BADGE													N/A	0.4206	0.0079	0.0079	0.0159	0.0159	0.0159	0.0079	0.0079	0.0079	0.0079	0.0079	0.0317	0.0079	0.0159	0.0317	0.0159	0.0317
	EGL														N/A	0.0317	0.0159	0.0079	0.0079	0.0159	0.0159	0.1508	0.0079	0.0079	0.0079	0.3095	0.1508	0.0079	0.0079	0.0079	0.2222
	Coreset															N/A	0.8413	0.0079	0.0079	0.0079	0.8413	0.0159	0.0079	0.0079	0.0079	0.0317	0.1508	0.0079	0.0079	0.0079	0.2222
MDNet	Random																N/A	0.0079	0.0079	0.0079	1	0.0079	0.0079	0.0079	0.0079	0.0159	0.0317	0.0079	0.0079	0.0079	0.1508
	Uncertainty																	N/A	1	1	0.0079	0.0079	0.3095	1	0.8413	0.0079	0.0079	0.2492	1	0.6905	0.0079
	BADGE																		N/A	1	0.0079	0.0079	0.3095	1	1	0.0079	0.0079	0.4206	0.8413	0.6905	0.0079
	EGL																			N/A	0.0079	0.0079	0.4206	1	0.6905	0.0079	0.0079	0.2222	1	0.8413	0.0079
	Coreset																				N/A	0.0079	0.0079	0.0079	0.0079	0.0159	0.0556	0.0079	0.0079	0.0079	0.1508
MAN	Random																					N/A	0.0079	0.0079	0.0079	0.1508	0.6905	0.0079	0.0079	0.0079	0.8413
	Uncertainty																						N/A	0.1508	0.2222	0.0079	0.0079	0.6905	0.0952	0.2222	0.0079
	BADGE																							N/A	0.5476	0.0079	0.0079	0.3095	0.8413	0.4206	0.0079
	EGL																								N/A	0.0079	0.0079	0.3095	1	0.6905	0.0079
	Coreset																									N/A	0.2222	0.0079	0.0079	0.0079	0.8413
CAN	Random																										N/A	0.0079	0.0079	0.0079	0.6905
	Uncertainty																											N/A	0.3095	0.2222	0.0079
	BADGE																												N/A	0.8413	0.0079
	EGL																													N/A	0.0079
	Coreset																														N/A

Appendix B

Supplementary Materials for Multi-Domain Learning from Insufficient Annotations

Supplementary Materials for Multi-Domain Learning from Insufficient Annotations in Chapter 4.

B.1 Statistical Analysis

All the results are analyzed using the Mann-Whitney U test (Mann and Whitney, 1947). Following multiple iterations of our experiment and subsequent statistical analysis, the corresponding p-value is presented. This value quantifies the statistical significance of the results obtained through our methodology.

The AULC values and corresponding p-values for comparisons under moderately insufficient labeled case (section 4.4.3.1) are shown in Table B.1.

The corresponding p-values for comparisons in the ablation study (section 4.4.4) are shown in Table B.2 and Table B.3.

The corresponding p-values for comparisons under active learning scenario (section 4.4.5) are shown from Table B.4 to Table B.6.

Table B.1: The AULC values and corresponding p-values of the Mann-Whitney U test for the moderately insufficient labeled case. The analysis is conducted on the AULC results.

	Amazon	MNIST-USPS	Office-Home	FDUMTL(4)	FDUMTL(16)	PACS
MAN	78.67(0.59)	79.33(1.28)	78.54(0.42)	75.47(0.96)	81.18(0.55)	88.99(0.52)
MDCL	79.61(0.57)	81.53(1.33)	80.50(0.41)	76.47(0.59)	81.93(0.45)	89.93(0.54)
p-value	0.018	0.005	0.000	0.040	0.041	0.004

Table B.2: The p-values resulting from the Mann-Whitney U test, applied to assess the components of MDCL on the **MNIST-USPS** dataset. This analysis specifically evaluates the accuracy performance when using 5% labeled training instances.

	MAN	MDCL (Inter)	MDCL (Intra)	MDCL
MAN	N/A	0.9138	0	0
MDCL (Inter)		N/A	0	0
MDCL (Intra)			N/A	0.607
MDCL				N/A

Table B.3: The p-values resulting from the Mann-Whitney U test, applied to assess the components of MDCL on the **PACS** dataset. This analysis specifically evaluates the accuracy performance when using 5% labeled training instances.

	MAN	MDCL (Inter)	MDCL (Intra)	MDCL
MAN	N/A	0.001	0.5205	0.0006
MDCL (Inter)		N/A	0.0022	0.2123
MDCL (Intra)			N/A	0.0008
MDCL				N/A

Table B.4: The p-values obtained from the Mann-Whitney U test, applied to evaluate the performance of MDCL in an active learning setting on the **Amazon** dataset. This analysis focuses on the corresponding Area Under the Learning Curve (AULC).

	MAN+Random	MAN+BvSB	MDCL+Random	MDCL+BvSB
MAN+Random	N/A	0.1049	0.0117	0.0002
MAN+BvSB		N/A	0.5737	0.0006
MDCL+Random			N/A	0.0019
MDCL+BvSB				N/A

Table B.5: The p-values obtained from the Mann-Whitney U test, applied to evaluate the performance of MDCL in an active learning setting on the **Office-Home** dataset. This analysis focuses on the corresponding Area Under the Learning Curve (AULC).

	MAN+Random	MAN+BvSB	MDCL+Random	MDCL+BvSB
MAN+Random	N/A	0.0002	0.0002	0.0002
MAN+BvSB		N/A	0.0003	0.0002
MDCL+Random			N/A	0.0002
MDCL+BvSB				N/A

Table B.6: The p-values obtained from the Mann-Whitney U test, applied to evaluate the performance of MDCL in an active learning setting on the **MNIST-USPS** dataset. This analysis focuses on the corresponding Area Under the Learning Curve (AULC).

	MAN+Random	MAN+BvSB	MDCL+Random	MDCL+BvSB
MAN+Random	N/A	0.0002	0.0379	0.0002
MAN+BvSB		N/A	0.0207	0.0086
MDCL+Random			N/A	0.0002
MDCL+BvSB				N/A

Appendix C

Supplementary Materials for Perturbation-Based Two-Stage Multi-Domain Active Learning

Supplementary Materials for Perturbation-Based Two-Stage Multi-Domain Active Learning in Chapter 5.

C.1 Statistical Analysis

All the AULC results from model-strategy pairs are analyzed using the Mann-Whitney U test (Mann and Whitney, 1947).

C.1.1 Statistical Analysis for Strategies

The p-values for each pair of model-strategy pairs are shown in table C.1 to C.3.

Table C.1: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **Amazon** dataset.

	Random	BvSB	BADGE	EGL	Coreset	P2S-MDAL
Random	N/A	0.0073	0.0036	0.1212	0.004	0.0022
BvSB		N/A	0.0376	0.0173	0.0002	0.0257
BADGE			N/A	0.0017	0.0002	0.3847
EGL				N/A	0.0003	0.0022
Coreset					N/A	0.0002
P2S-MDAL						N/A

Table C.2: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **COIL** dataset.

	Random	BvSB	BADGE	EGL	Coreset	P2S-MDAL
Random	N/A	0.0002	0.0002	0.0013	0.0002	0.0002
BvSB		N/A	0.0757	0.0493	0.0022	0.0091
BADGE			N/A	0.0022	0.0695	0.2411
EGL				N/A	0.0002	0.0008
Coreset					N/A	0.4725
P2S-MDAL						N/A

Table C.3: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results on **FDUMTL** dataset.

	Random	BvSB	BADGE	EGL	Coreset	P2S-MDAL
Random	N/A	0.0028	0.0019	0.4619	0.0074	0.0009
BvSB		N/A	0.9581	0.0054	0.1036	0.4948
BADGE			N/A	0.0054	0.5635	0.1278
EGL				N/A	0.0136	0.0009
Coreset					N/A	0.0356
P2S-MDAL						N/A

C.1.2 Statistical Analysis for Ablation Study

The p-values for the comparison of model components are shown in table C.4 and C.5.

Table C.4: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results from **ablation study**.

	w/o_both	w/o_perturb	w/o_region	P2S-MDAL
w/o_both	N/A	0.0009	0.0406	0.0009
w/o_perturb		N/A	0.0181	0.6365
w/o_region			N/A	0.0181
P2S-MDAL				N/A

Table C.5: The p-values of the Mann-Whitney U test for the models. The analysis is conducted on the AULC results from **perturbation analysis**.

	2S_Center	2S_BvSB	2S_EGL	P2S-MDAL
2S_Center	N/A	0.372	0.7132	0.7929
2S_BvSB		N/A	1	0.2701
2S_EGL			N/A	0.1278
P2S-MDAL				N/A

References

[1] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". *arXiv preprint arXiv:1803.08375* (2018).

[2] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. "Contour detection and hierarchical image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2010), pp. 898–916.

[3] Jordan T. Ash, Chicheng Zhang, et al. "Deep batch active learning by diverse, uncertain gradient lower bounds". *Proceedings of the 8th International Conference on Learning Representations*. 2020.

[4] Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. "Gone Fishing: Neural Active Learning with Fisher Embeddings". *Proceedings of the Annual Conference on Neural Information Processing Systems 2021*. 2021, pp. 8927–8939.

[5] Ricardo Barata et al. "Active learning for online training in imbalanced data streams under cold start". *arXiv preprint arXiv:2107.07724* (2021).

[6] Nathan Beck et al. "Effective Evaluation of Deep Active Learning on Image Classification Tasks". *arXiv preprint arXiv:2106.15324* (2021).

[7] Javad Zolfaghari Bengar, Joost van de Weijer, Bartlomiej Twardowski, and Bogdan C. Raducanu. "Reducing Label Effort: Self-Supervised meets Active Learning".

Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops. 2021, pp. 1631–1639.

[8] Erdem Biyik, Kenneth Wang, Nima Anari, and Dorsa Sadigh. "Batch Active Learning Using Determinantal Point Processes". *arXiv preprint arXiv:1906.07975* (2019).

[9] Olivier Borkowski et al. "Large scale active-learning-guided exploration for in vitro protein production optimization". *Nature communications* 11.1 (2020), p. 1872.

[10] Konstantinos Bousmalis et al. "Domain Separation Networks". *Proceedings of the Annual Conference on Neural Information Processing Systems 2016*. 2016, pp. 343–351.

[11] Tom B. Brown et al. "Language Models are Few-Shot Learners". *Proceedings of the Annual Conference on Neural Information Processing Systems 2020*. 2020.

[12] Davide Cacciarelli and Murat Kulahci. "A survey on online active learning". *arXiv preprint arXiv:2302.08893* (2023).

[13] Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. "Visual Transformer for Task-aware Active Learning". *arXiv preprint arXiv:2106.03801* abs/2106.03801 (2021).

[14] Augustin Cauchy et al. "Méthode générale pour la résolution des systemes d'équations simultanées". *Comp. Rend. Sci. Paris* 25.1847 (1847), pp. 536–538.

[15] Ji Chang et al. "Active Domain Adaptation With Application to Intelligent Logging Lithology Identification". *IEEE Transactions on Cybernetics* 52.8 (2022), pp. 8073–8087.

[16] Rita Chattopadhyay et al. "Joint transfer and batch-mode active learning". *Proceedings of the 30th International Conference on Machine Learning*. 2013, pp. 253–261.

[17] Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. "Marginalized denoising autoencoders for domain adaptation" (2012).

[18] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. "A Simple Framework for Contrastive Learning of Visual Representations". *Proceedings of the 37th International Conference on Machine Learning*. 2020, pp. 1597–1607.

[19] Xilun Chen and Claire Cardie. "Multinomial adversarial networks for multi-domain text classification". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2018, pp. 1226–1240.

[20] Gui Citovsky et al. "Batch Active Learning at Scale". *Proceedings of the Annual Conference on Neural Information Processing Systems 2021*. 2021, pp. 11933–11944.

[21] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. "Active Learning with Statistical Models". *Proceedings of the Annual Conference on Neural Information Processing Systems 1994*. 1994, pp. 705–712.

[22] Imre Csiszár. "I-divergence geometry of probability distributions and minimization problems". *The annals of probability* (1975), pp. 146–158.

[23] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. "Boosting for transfer learning". *Machine Learning, Proceedings of the 24th International Conference*. 2007, pp. 193–200.

[24] Zeyu Dai, Shengcai Liu, Qing Li, and Ke Tang. "Saliency Attack: Towards Imperceptible Black-Box Adversarial Attack". *ACM Transactions on Intelligent Systems and Technology* 14.3 (2023).

[25] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. "Active multi-kernel domain adaptation for hyperspectral image classification". *Pattern Recognition* 77 (2018), pp. 306–315.

[26] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. "Solving the Multiple Instance Problem with Axis-Parallel Rectangles". *Artificial Intelligence* 89.1-2 (1997), pp. 31–71.

[27] Xiaoyu Ding et al. "Active learning for drug design: a case study on the plasma exposure of orally administered drugs". *Journal of Medicinal Chemistry* 64.22 (2021), pp. 16838–16853.

[28] Jeff Donahue et al. "DeCAF: a deep convolutional activation feature for generic visual Recognition". *Proceedings of the 31th International Conference on Machine Learning*. 2014, pp. 647–655.

[29] Mark Dredze and Koby Crammer. "Online methods for multi-domain learning and adaptation". *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 2008, pp. 689–697.

[30] Melanie Ducoffe and Frédéric Precioso. "Adversarial Active Learning for Deep Networks: a Margin Based Approach". *arXiv preprint arXiv:1802.09841* (2018).

[31] Melanie Ducoffe and Frédéric Precioso. "Adversarial Active Learning for Deep Networks: a Margin Based Approach". *arXiv preprint arXiv:1802.09841* (2018).

[32] Sayna Ebrahimi, William Gan, Kamyar Salahi, and Trevor Darrell. "Minimax Active Learning". *arXiv preprint arXiv:2012.10467* (2020).

[33] Mehdi Elahi, Francesco Ricci, and Neil Rubens. "A survey of active learning in collaborative filtering recommender systems". *Computer Science Review* 20 (2016), pp. 29–50.

[34] Jesper E. van Engelen and Holger H. Hoos. "A survey on semi-supervised learning". *Machine Learning* 109.2 (2020), pp. 373–440.

[35] Meng Fang, Yuan Li, and Trevor Cohn. "Learning how to Active Learn: A Deep Reinforcement Learning Approach". *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 595–605.

[36] Meng Fang, Jie Yin, and Xingquan Zhu. "Knowledge Transfer for Multi-labeler Active Learning". *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013*. Vol. 8188. 2013, pp. 273–288.

[37] Steven Y Feng, Varun Gangal, et al. "A Survey of Data Augmentation Approaches for NLP". *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021, pp. 968–988.

[38] Zeyu Feng, Chang Xu, and Dacheng Tao. "Self-supervised representation learning from multi-domain data". *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3244–3254.

[39] Eugene Yujun Fu et al. "Exploiting Active Learning in Novel Refractive Error Detection with Smartphones". *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 2775–2783.

[40] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep Bayesian Active Learning with Image Data". *Proceedings of the 34th International Conference on Machine Learning*. 2017, pp. 1183–1192.

[41] Yaroslav Ganin et al. "Domain-adversarial training of neural networks". *The Journal of Machine Learning Research* 17 (2016), 59:1–59:35.

[42] Mingfei Gao, Zizhao Zhang, et al. "Consistency-Based Semi-supervised Active Learning: Towards Minimizing Labeling Cost". *Proceedings of the 2020 Computer Vision European Conference*. 2020, pp. 510–526.

[43] Ruijiang Gao and Maytal Saar-Tsechansky. "Cost-Accuracy Aware Adaptive Labeling for Active Learning". *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. 2020, pp. 2569–2576.

[44] Daniel Gissin and Shai Shalev-Shwartz. "Discriminative Active Learning". *arXiv preprint arXiv:1907.06347* (2019).

[45] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. "Geodesic flow kernel for unsupervised domain adaptation". *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 2066–2073.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[47] Rui He, Zeyu Dai, Shan He, and Ke Tang. "Perturbation-Based Two-Stage Multi-Domain Active Learning". *arXiv preprint arXiv:2306.10700* (2023).

[48] Rui He, Shengcai Liu, Shan He, and Ke Tang. "Multi-Domain Active Learning: Literature Review and Comparative Study". *IEEE Transactions on Emerging Topics in Computational Intelligence* 7.3 (2023), pp. 791–804.

[49] Rui He, Shengcai Liu, Jiahao Wu, et al. "Multi-Domain Learning From Insufficient Annotations". *arXiv preprint arXiv:2305.02757* (2023).

[50] Geoffrey E. Hinton, Nitish Srivastava, et al. "Improving neural networks by preventing co-adaptation of feature detectors". *arXiv preprint arXiv:1207.0580* (2012).

[51] Geoffrey Hinton, Li Deng, et al. "Deep neural networks for acoustic modeling in speech recognition". *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97.

[52] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. "Meta-Learning in Neural Networks: A Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2022), pp. 5149–5169.

[53] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. "Bayesian Active Learning for Classification and Preference Learning". *arXiv preprint arXiv:1112.5745* (2011).

[54] Wei-Ning Hsu and Hsuan-Tien Lin. "Active Learning by Learning". *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2015, pp. 2659–2665.

[55] Chao Huang, Hu Han, et al. "3D U^2-Net: A 3D Universal U-Net for Multi-domain Medical Image Segmentation". *Proceedings of Medical Image Computing and Computer Assisted Intervention - MICCAI*. 2019, pp. 291–299.

[56] Sheng-Jun Huang, Jia-Lve Chen, Xin Mu, and Zhi-Hua Zhou. "Cost-Effective Active Learning from Diverse Labelers". *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 2017, pp. 1879–1885.

[57] Sheng-Jun Huang and Songcan Chen. "Transfer learning with active queries from source domain". *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 2016, pp. 1592–1598.

[58] Zenan Huang, Jun Wen, et al. "Discriminative radial domain adaptation". *IEEE Transactions on Image Processing* 32 (2023), pp. 1419–1431.

[59] Jonathan J. Hull. "A Database for Handwritten Text Recognition Research". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.5 (1994), pp. 550–554.

[60] Hal Daumé III. "Frustratingly easy domain adaptation". *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. 2007.

[61] Jing Jiang and ChengXiang Zhai. "Instance Weighting for Domain Adaptation in NLP". *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. 2007.

[62] Wei Jin et al. "Condensing Graphs via One-Step Gradient Matching". *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2022, pp. 720–730.

[63] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. "Multi-class active learning for image classification". *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2009, pp. 2372–2379.

[64] Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen. "Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks". *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 680–688.

[65] Prannay Khosla et al. "Supervised Contrastive Learning". *Proceedings of the Annual Conference on Neural Information Processing Systems 2020*. 2020.

[66] Yoon-Yeong Kim, Kyungwoo Song, JoonHo Jang, and Il-chul Moon. "LADA: Look-Ahead Data Acquisition via Augmentation for Deep Active Learning". *Proceedings of the Annual Conference on Neural Information Processing Systems 2021*. 2021, pp. 22919–22930.

[67] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. "BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning". *Proceedings of the Annual Conference on Neural Information Processing Systems 2019*. 2019, pp. 7024–7035.

[68] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. "Learning Active Learning from Data". *Proceedings of the Annual Conference on Neural Information Processing Systems 2017*. 2017, pp. 4225–4235.

[69] Ranganath Krishnan et al. "Mitigating Sampling Bias and Improving Robustness in Active Learning". *arXiv preprint arXiv:2109.06321* (2021).

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". *Advances in neural information processing systems*. Vol. 25. 2012, pp. 1097–1105.

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

[72] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[73] Peter Lee, Sebastien Bubeck, and Joseph Petro. "Benefits, limits, and risks of GPT-4 as an AI chatbot for medicine". *New England Journal of Medicine* 388.13 (2023), pp. 1233–1239.

[74] David D. Lewis and Jason Catlett. "Heterogeneous Uncertainty Sampling for Supervised Learning". *Machine Learning, Proceedings of the 11th International Conference*. 1994, pp. 148–156.

[75] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. "Deeper, Broader and Artier Domain Generalization". *Proceedings of the 2017 IEEE International Conference on Computer Vision*. 2017, pp. 5543–5551.

[76] Kai Li, Chang Liu, et al. "Ecacl: A holistic framework for semi-supervised domain adaptation". *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops*. 2021, pp. 8578–8587.

[77] Lianghao Li, Xiaoming Jin, Sinno Jialin Pan, and Jian-Tao Sun. "Multi-domain active learning for text classification". *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2012, pp. 1086–1094.

[78] Yingming Li, Ming Yang, and Zhongfei Zhang. "A Survey of Multi-View Representation Learning". *IEEE Transactions on Knowledge and Data Engineering* 31.10 (2019), pp. 1863–1883.

[79] Yitong Li, Timothy Baldwin, and Trevor Cohn. "Semi-supervised stochastic multi-domain learning using variational inference". *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 2019, pp. 1923–1934.

[80] Yunsheng Li and Nuno Vasconcelos. "Efficient multi-domain learning by covariance normalization". *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5424–5433.

[81] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. "Transferable Adversarial Training: A General Approach to Adapting Deep Classifiers". *Proceedings of the 36th International Conference on Machine Learning*. 2019, pp. 4013–4022.

[82] Ming Liu, Wray L. Buntine, and Gholamreza Haffari. "Learning How to Actively Learn: A Deep Imitation Learning Approach". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018, pp. 1874–1883.

[83] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. "Adversarial Multi-task Learning for Text Classification". *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, pp. 1–10.

[84] Shengcai Liu, Ke Tang, and Xin Yao. "Generative Adversarial Construction of Parallel Portfolios". *IEEE Transactions on Cybernetics* 52.2 (2022), pp. 784–795.

[85] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. "Learning transferable features with deep adaptation networks". *Proceedings of the 32nd International Conference on Machine Learning*. 2015, pp. 97–105.

[86] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. "Conditional adversarial domain adaptation". *Proceedings of the Annual Conference on Neural Information Processing Systems 2018*. 2018, pp. 1647–1657.

[87] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. "Deep transfer learning with joint adaptation networks". *Proceedings of the 34th International Conference on Machine Learning*. 2017, pp. 2208–2217.

[88] Ning Lu, Shengcai Liu, Rui He, and Ke Tang. "Large Language Models can be Guided to Evade AI-Generated Text Detection". *arXiv preprint arXiv:2305.10847* (2023).

[89] Henry B Mann and Donald R Whitney. "On a test of whether one of two random variables is stochastically larger than the other". *The annals of mathematical statistics* (1947), pp. 50–60.

[90] Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. "Active Learning Principles for In-Context Learning with Large Language Models". *arXiv preprint arXiv:2305.14264* (2021).

[91] Katerina Margatina, Giorgos Vernikos, LoıHERE!HERE!c Barrault, and Nikolaos Aletras. "Active Learning by Acquiring Contrastive Examples". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 650–663.

[92] Idriss Mghabbar and Pirashanth Ratnamogan. "Building a Multi-Domain Neural Machine Translation Model Using Knowledge Distillation". *Proceedings of 24th European Conference on Artificial Intelligence*. Vol. 325. 2020, pp. 2116–2123.

[93] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. "Unified deep supervised domain adaptation and generalization". *Proceedings of the 2017 IEEE International Conference on Computer Vision*. 2017, pp. 5716–5726.

[94] Prateek Munjal et al. "Towards Robust and Reproducible Active Learning using Neural Networks". *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 223–232.

[95] David Nadeau and Satoshi Sekine. "A survey of named entity recognition and classification". *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[96] Hyeonseob Nam and Bohyung Han. "Learning multi-domain convolutional neural networks for visual tracking". *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4293–4302.

[97] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. "Columbia object image library (coil-20)". *Technical Report CUCS-005-96* (1996).

[98] Hieu Tat Nguyen and Arnold W. M. Smeulders. "Active learning using pre-clustering". *Machine Learning, Proceedings of the 21st International Conference*. Vol. 69. 2004.

[99] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". *arXiv preprint arXiv:1807.03748* (2018).

[100] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. "Domain adaptation via transfer component analysis". *IEEE Transactions on Neural Networks* 22.2 (2011), pp. 199–210.

[101] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.

[102] Yingwei Pan, Ting Yao, et al. "Transferrable Prototypical Networks for Unsupervised Domain Adaptation". *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2239–2247.

[103] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. "Multi-adversarial domain adaptation". *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 2018, pp. 3934–3941.

[104] Fengchao Peng, Chao Wang, Jianzhuang Liu, and Zhen Yang. "Active Learning for Lane Detection: A Knowledge Distillation Approach". *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15132–15141.

[105] Sylvestre Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. "Efficient parametrization of multi-domain deep neural networks". *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8119–8127.

[106] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. "Learning multiple visual domains with residual adapters". *Proceedings of the Annual Conference on Neural Information Processing Systems 2017*. 2017, pp. 506–516.

[107] Pengzhen Ren et al. "A Survey of Deep Active Learning". *ACM Computing Surveys* 54.9 (2022), 180:1–180:40.

[108] Nicholas Roy and Andrew McCallum. "Toward Optimal Active Learning through Sampling Estimation of Error Reduction". *Proceedings of the 18th International Conference on Machine Learning*. 2001, pp. 441–448.

[109] Virginia R. de Sa. "Learning Classification with Unlabeled Data". *Proceedings of the Annual Conference on Neural Information Processing Systems 1993*. 1993, pp. 112–119.

[110] Stephan R Sain. *The nature of statistical learning theory*. 1996.

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

[112] Peter Peter Samoaa et al. "A Unified Active Learning Framework for Annotating Graph Data with Application to Software Source Code Performance Prediction". *arXiv preprint arXiv:2304.13032* (2023).

[113] Andrew I. Schein and Lyle H. Ungar. "Active learning for logistic regression: an evaluation". *Machine Learning* 68.3 (2007), pp. 235–265.

[114] Christopher Schröder, Andreas Niekler, and Martin Potthast. "Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers". *Findings of the Association for Computational Linguistics: ACL 2022*. 2022, pp. 2194–2203.

[115] Ozan Sener and Silvio Savarese. "Active Learning for Convolutional Neural Networks: A Core-Set Approach". *Proceedings of the 6th International Conference on Learning Representations*. 2018.

[116] Seungmin Seo, Donghyun Kim, Youbin Ahn, and Kyong-Ho Lee. "Active Learning on Pre-trained Language Model with Task-Independent Triplet Loss". *Proceedings of Thirty-Sixth AAAI Conference on Artificial Intelligence*. 2022, pp. 11276–11284.

[117] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.

[118] Burr Settles and Mark Craven. "An analysis of active learning strategies for sequence labeling tasks". *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 2008, pp. 1070–1079.

[119] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. "Query by Committee". *Proceedings of the 5th Annual ACM Conference on Computational Learning Theory*. 1992, pp. 287–294.

[120] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". *Journal of big data* 6.1 (2019), pp. 1–48.

[121] Connor Shorten and Taghi M. Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". *Journal of Big Data* 6 (2019), p. 60.

[122] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. "Deep Active Learning: Unified and Principled Method for Query and Training". *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. 2020, pp. 1308–1318.

[123] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. "Rethinking deep active learning: Using unlabeled data at model training". *Proceedings of the 25th International Conference on Pattern Recognition*. 2020, pp. 1220–1227.

[124] Ankit Singh. "Clda: Contrastive learning for semi-supervised domain adaptation". *Proceedings of the Annual Conference on Neural Information Processing Systems 2021*. 2021, pp. 5089–5101.

[125] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. "Variational Adversarial Active Learning". *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5971–5980.

[126] Ard Snijders, Douwe Kiela, and Katerina Margatina. "Investigating Multi-source Active Learning for Natural Language Inference". *arXiv preprint arXiv:2302.06976* (2023).

[127] Jong-Chyi Su et al. "Active adversarial domain adaptation". *Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 728–737.

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

[129] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[130] Rinyoichi Takezoe et al. "Deep Active Learning for Computer Vision: Past and Future". *arXiv preprint arXiv:2211.14819* (2022).

[131] Alex Tamkin et al. "Active Learning Helps Pretrained Models Learn the Intended Task". *arXiv preprint arXiv:2204.08491* (2022).

[132] Ke Tang, Shengcai Liu, Peng Yang, and Xin Yao. "Few-Shots Parallel Algorithm Portfolio Construction via Co-Evolution". *IEEE Transactions on Evolutionary Computation* 25.3 (2021), pp. 595–607.

[133] Ying-Peng Tang and Sheng-Jun Huang. "Self-Paced Active Learning: Query the Right Thing at the Right Time". *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. 2019, pp. 5117–5124.

[134] Korawat Tanwisuth et al. "A prototype-oriented framework for unsupervised domain adaptation". *Proceedings of the Annual Conference on Neural Information Processing Systems 2021*. 2021, pp. 17194–17208.

[135] Annalisa T. Taylor, Thomas A. Berrueta, and Todd D. Murphey. "Active Learning in Robotics: A Review of Control Principles". *arXiv preprint arXiv:2106.13697* (2022).

[136] Mario Teixeira Parente et al. "Active learning-assisted neutron spectroscopy with log-Gaussian processes". *Nature Communications* 14.1 (2023), p. 2246.

[137] Simon Tong. "Active learning: theory and applications". PhD thesis. Stanford University, 2001.

[138] Toan Tran, Thanh-Toan Do, Ian D. Reid, and Gustavo Carneiro. "Bayesian Generative Active Deep Learning". *Proceedings of the 36th International Conference on Machine Learning*. 2019, pp. 6295–6304.

[139] Devis Tuia et al. "A survey of active learning algorithms for supervised remote sensing image classification". *arXiv preprint arXiv:2104.07784* (2021).

[140] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. "Adversarial discriminative domain adaptation". *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2962–2971.

[141] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. "Deep Hashing Network for Unsupervised Domain Adaptation". *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5385–5394.

[142] Vincent Vercruyssen, Lorenzo Perini, Wannes Meert, and Jesse Davis. "Multi-domain Active Learning for Semi-supervised Anomaly Detection". *ECML/PKDD 2022 published proceedings* (2022).

[143] Luués Nunes Vicente and Paul H. Calamai. "Bilevel and multilevel programming: A bibliography review". *Journal of Global optimization* 5.3 (1994), pp. 291–306.

[144] Jindong Wang, Cuiling Lan, et al. "Generalizing to Unseen Domains: A Survey on Domain Generalization". *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. 2021, pp. 4627–4635.

[145] Keze Wang, Dongyu Zhang, et al. "Cost-Effective Active Learning for Deep Image Classification". *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2017), pp. 2591–2600.

[146] Shuo Wang, Yuexiang Li, et al. "Dual Adversarial Network for Deep Active Learning". *Proceedings of the 2020 Computer Vision European Conference*. Vol. 12369. 2020, pp. 680–696.

[147] Tongzhou Wang and Phillip Isola. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". *Proceedings of the 37th International Conference on Machine Learning*. 2020, pp. 9929–9939.

[148] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. "Generalizing from a Few Examples: A Survey on Few-shot Learning". *ACM Computing Surveys* 53.3 (2021), 63:1–63:34.

[149] Jiahao Wu, Wenqi Fan, Jingfan Chen, et al. "Disentangled Contrastive Learning for Social Recommendation". *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, pp. 4570–4574.

[150] Jiahao Wu, Wenqi Fan, Rui He, et al. "Dataset Condensation for Recommendation". Unpublished Work. 2023.

[151] Pengcheng Wu and Thomas G. Dietterich. "Improving SVM accuracy by training on auxiliary data sources". *Machine Learning, Proceedings of the 21st International Conference*. 2004.

[152] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. "Conditional Adversarial Networks for Multi-Domain Text Classification". *arXiv preprint arXiv:2102.10176* (2021).

[153] Jin Xiao, Shuhang Gu, and Lei Zhang. "Multi-domain learning for accurate and few-shot color constancy". *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3255–3264.

[154] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. "Learning deep feature representations with domain guided dropout for person re-identification". *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1249–1258.

[155] Weilin Xu, David Evans, and Yanjun Qi. "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks". *25th Annual Network and Distributed System Security Symposium*. Vol. 325. 2018, pp. 2116–2123.

[156] Zhao Xu, Kai Yu, et al. "Representative Sampling for Text Classification Using Support Vector Machines". *Proceedings of the 25th European Conference on IR Research*. Vol. 2633. 2003, pp. 393–407.

[157] Yan Yan, Rómer Rosales, Glenn Fung, and Jennifer G. Dy. "Active Learning from Crowds". *Proceedings of the 28th International Conference on Machine Learning, ICML*. 2011, pp. 1161–1168.

[158] Zizheng Yan, Yushuang Wu, et al. "Multi-level consistency learning for semi-supervised domain adaptation". *arXiv preprint arXiv:2205.04066* (2022).

[159] Jing Yao, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. "Looking Back on the Past: Active Learning with Historical Evaluation Results". *IEEE Transactions on Knowledge and Data Engineering* (2020), pp. 1–1.

[160] Tianxiang Yin, Ningzhong Liu, and Han Sun. "Self-paced active learning for deep CNNs via effective loss function". *Neurocomputing* 424 (2021), pp. 1–8.

[161] Donggeun Yoo and In So Kweon. "Learning Loss for Active Learning". *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 93–102.

[162] Junkun Yuan, Xu Ma, et al. "Label-Efficient Domain Generalization via Collaborative Exploration and Generalization". *Proceedings of The 30th ACM International Conference on Multimedia*. 2022, pp. 2361–2370.

[163] Michelle Yuan, Hsuan-Tien Lin, and Jordan L. Boyd-Graber. "Cold-start Active Learning through Self-supervised Language Modeling". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 2020, pp. 7935–7948.

[164] Xueying Zhan, Huan Liu, Qing Li, and Antoni B. Chan. "A Comparative Survey: Benchmarking for Pool-based Active Learning". *Proceedings of the 30th International Joint Conference on Artificial Intelligenc*. 2021.

[165] Xueying Zhan, Qingzhong Wang, et al. "A Comparative Survey of Deep Active Learning". *arXiv preprint arXiv:2203.13450* (2022).

[166] Chicheng Zhang and Kamalika Chaudhuri. "Active Learning from Weak and Strong Labelers". *Proceedings of the Annual Conference on Neural Information Processing Systems 2015*. 2015, pp. 703–711.

[167] Min-Ling Zhang and Zhi-Hua Zhou. "A Review on Multi-Label Learning Algorithms". *IEEE Transactions on Knowledge and Data Engineering* 26.8 (2014), pp. 1819–1837.

[168] Ye Zhang, Matthew Lease, and Byron C. Wallace. "Active discriminative text representation learning". *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 2017, pp. 3386–3392.

[169] Yu Zhang, Peter Tiňo, Ales Leonardis, and Ke Tang. "A Survey on Neural Network Interpretability". *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742.

[170] Yu Zhang and Qiang Yang. "A Survey on Multi-Task Learning". *IEEE Transactions on Knowledge and Data Engineering* (2021).

[171] Zhisong Zhang, Emma Strubell, and Eduard H. Hovy. "A Survey of Active Learning for Natural Language Processing". *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 6166–6190.

[172] Zihan Zhang, Xiaoming Jin, et al. "Multi-domain active learning for recommendation". *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. 2016, pp. 2358–2364.

[173] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. "Dataset Condensation with Gradient Matching". *Proceedings of the 9th International Conference on Learning Representations*.

[174] Guang Zhao, Edward R. Dougherty, et al. "Uncertainty-aware Active Learning for Optimal Bayesian Classifier". *Proceedings of the 9th International Conference on Learning Representations*. 2021.

[175] Fedor Zhdanov. "Diverse mini-batch Active Learning". *arXiv preprint arXiv:1901.05954* (2019).

[176] Hao Zheng et al. "Hierarchical Self-supervised Learning for Medical Image Segmentation Based on Multi-domain Data Aggregation". *Proceedings of Medical Image Computing and Computer Assisted Intervention - MICCAI*. 2021, pp. 622–632.