UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

DEEP NEURAL NETWORKS FOR MULTI-SOURCE TRANSFER LEARNING

by

Keqiuyin Li

A Thesis Submitted
in Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2022

Certificate of Original Authorship

I, Keqiuyin Li, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 17 Oct, 2022

ABSTRACT

Transfer learning is gaining incredible attention due to its ability to leverage previously acquired knowledge from source domain to assist in completing a task in a similar target domain. Many existing transfer learning methods deal with single source and single target transfer learning, but rarely consider the fact that information from a single source can be inadequate to a target domain and there can be multiple source domains. Few multi-source domain adaptations methods adapt all source and target data into a same latent feature space. However, domain shifts can be found among source domains and between each pair of source and target domains, thus, the model fitting all domains well may not exist. In addition, most transfer learning methods assume that the source and target domains share the same label space. But in practice, the source domain(s) sharing the same label space with the target domain may never be found. Third, data privacy and security are being magnificently conspicuous in real-world applications, which means the traditional transfer learning relying on data matching can trigger privacy concerns.

To solve the above-mentioned problems, this thesis develops a series of methods to tackle transfer learning with multiple source domains. Knowledge transfer with and without source data are explored under both homogeneous and heterogeneous label space settings.

To tackle knowledge transfer from multiple source domains, and measure contributions of source domains, multi-source contribution learning and dynamic classifier alignment methods are developed. In multi-source contribution learning method, the similarities and diversities of domains are learned simultaneously by extracting multi-view features. One view represents common features (similarities) among all domains. Other views represent different characteristics (diversities) in a target domain, in which each characteristic is expressed by features extracted in a source domain. Then multi-level distribution matching is employed to improve the transferability of latent features, aiming to reduce misclassification of boundary samples

by maximizing discrepancy between different classes and minimizing discrepancy between the same classes. Concurrently, when completing a target task by combining source predictions, instead of averaging source predictions or weighting sources using normalized similarities, the original weights learned by normalizing similarities between source and target domains are adjusted using pseudo target labels to increase the disparities of weight values, which is desired to improve the performance of the final target predictor if the predictions of sources exist significant difference.

In dynamic classifier alignment method, it aligns classifiers driven from multi-view features via a sample-wise automatic way. As proposed, both the importance of each view and the contribution of each source domain are investigated. To determine the important degrees of multiple views, an importance learning function is built by generating an auxiliary classifier. To learn the source combination parameters, a domain discriminator is developed to estimate the probability of a sample belonging to multiple source domains. Meanwhile, a self-training strategy is proposed to enhance the cross-domain ability of source classifiers with the assistance of pseudo target labels.

To learn similarity of source and target domains to define what to transfer, sample and source distillation method is proposed. It develops a two-step selective strategy to distill source samples and define the importance of source domains. To distill samples, the pseudo-labeled target domain is constructed to learn a series of category classifiers to identify transfer and inefficient source samples. To rank domains, a domain discriminator, which returns the degrees of a target sample belonging to the source domains, is developed based on selected transfer samples. Using the selected samples and ranked domains, transfer between the source and target domains is achieved by adapting multi-level distribution in a latent feature space. Furthermore, to explore more usable target information which is expected to enhance the cross-domain ability of source predictors, an enhancement mechanism is built by matching selected pseudo-labeled and unlabeled target samples. The degrees learned by the domain discriminator are finally employed to combine source predictors when predicting the target task.

To address transfer learning without the access to source data, generally auxiliary model training method is explored. The proposed method fits the source models to the target domain via fine-tuning under the supervision of pseudo target labels rather than matching data distributions. To collect high-quality initial pseudo target labels, both specific and generally auxiliary source models are pre-trained to improve the generality across domains of source models based on auxiliary learning, where source contributions are determined using an automatic way. Besides, the generally auxiliary model can take the benefit of sharing knowledge from multiple source domains without sharing data. Going further, it introduces a class balanced coefficient of each category based on the number of samples to reduce the misclassification caused by data imbalance.

To deal with soft information in transfer learning, fuzzy rule-based deep neural network is proposed to achieve multi-source data-free transfer learning. It takes advantage of a fuzzy system to handle data uncertainty in domain adaptation without source data. To learn source private models with high generality, which is important to collect low noisy pseudo target labels, auxiliary tasks are designed by jointly training source models from multiple domains which share source parameters and fuzzy rules while protecting source data. To transfer fuzzy rules and fit source private parameters to the target domain, self-supervised learning and anchor-based alignment are built to force target data to source feature spaces.

To handle transfer learning where source and target domains have unshared label space, partial and open-set transfer learning with generally auxiliary model training and fuzzy rules are explored under source-free setting. Universal transfer learning method is developed under multi-source-absent setting. In partial source-free transfer learning, a selection method is built to remove source samples from unshared categories, which is expected to eliminate the negative transfer resulting from the source outliers. In open-set transfer learning, a threshold generated from the predictions of the pre-trained source models is defined to identify the unknown target samples, aiming to eliminate the pseudo label noise caused by introducing unshared target samples.

In universal transfer learning, a unified learning model is proposed. The proposed method designs a module that can transfer knowledge from multi-source domains with both homogeneous and heterogeneous label spaces in universal scenario without accessing the source data. To classify known target classes, source anchors are generated to build data-matching between source and target domains via a contrastive method. In addition, class center consistency is adopted to distinguish source private samples when pseudo-labeling the target data to reduce label noise. To detect unknown classes, a clustering strategy which combines global and source local entropy assumptions is adopted to recognize the known and unknown target samples. By removing source private classes and target unknown samples, highly confident target samples are collected to self-supervise the adaptation of the pre-trained source model. At the same time, constraints enlarging the distance among target known classes and between the known and unknown samples are applied based on the pseudo-labels to enhance the performance of the proposed model.

Dedication

To myself and my family.

Acknowledgements

I would like to express my deepest gratitude to my principal supervisor Distinguished Professor Jie Lu, and my co-supervisors A/Professor Guangquan Zhang and Dr. Hua Zuo, who generously provided guidance and feedback during the past three years. Without their expertise and guidance, I could not have made it through my PhD degree.

Distinguished Professor Jie Lu played a decisive role in guiding me to develop research work as an academic researcher. She placed unconditional support in my research interests and had great patience to enlighten me to overcome the difficulties I encountered. Her invaluable suggestions benefited me in study and life. As an international student living alone in a foreign country, I felt honoured enough to have her as my supervisor, who loves students like her own children and respects our feelings and ideas. What I have learned form her will be great treasure in my life. A/Professor Guangquan Zhang gave me constructive advice on how to start research efficiently. His wisdom insights and expertise avoided me from losing my way in research. Without his critical comments, I would waste much time at the beginning to advance the first step of the research topic. Dr. Hua Zuo raised helpful points in our discussion which helped me completing my research. Her comments helped a lot to enrich the content and improve the quality of all my papers. I extremely appreciate to them for their help and support.

I am also grateful to the University of Technology Sydney and Australian Research Council (grand under FL190100149), who provided financial support for my research. Thanks to that, I had got the opportunity to experience this exciting journey as a PhD student in University of Technology Sydney, which absolutely expanded my knowledge and widened my view.

I had the pleasure of working with all members of Decision System and e-Service Intelligent Lab and Australian Artificial Intelligence Institute. They created positive study and work environment to keep physical and mental health for everyone. I would like to acknowledge honorary member Dr. Dianshuang Wu, Dr Feng Liu and Dr Yiliao Song, who helped me in both life and research. Dr. Yi Zhang provided valuable comments on reference management which benefited my research. Dr. Bin Wang helped me in building software environment. I would also like to thank Dr. Junyu Xuan, Dr. Qian Zhang, Dr. Anjin Liu, Dr. Zhen Fang, Dr. Ruxia Liang, Dr. Xiaoya Che, Tianyu Liu, Kun Wang and Mengjia Wu who shared their opinions and comments. Many thanks to Jim Howes who gave beneficial feedback in English to improve my writing. I also thank Camila Cremonese, Robyn Barden and Lily Qian, who helped in managing proofreading.

Thanks should also go to my parents Mr. Shixuan Li, Mrs. Meirong Guo and my younger brother Mr. Kesen Li. Their belief in me has kept my spirits and motivation high. Lastly, I'd like to mention other family members Jixiang and Ruyi, the Chinese rural dogs, and Kumar, the border collie, for all the entertainment and emotional support.

Keqiuyin Li
Sydney, Australia, 2022.

List of Publications

Journal Papers

J-1. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Multi-source contribution learning for domain adaptation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, 2022, pp 5293 - 5307.[A*; Q1]

J-2. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Dynamic classifier alignment for unsupervised multi-source domain adaptation,” *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2022.3144423, 2022. [A*; Q1]

J-3. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Multi-domain adaptation with sample and source distillation,” *IEEE Transactions on Cybernetics*, DOI 10.1109/TCYB.2023.3236008. [A; Q1]

J-4. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Source-free multi-domain adaptation with fuzzy rule based deep neural networks,” *IEEE Transactions on Fuzzy System.* Under review. [A*; Q1]

J-5. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Unified Learning for Source-Absent Universal Multi-Domain Adaptation,” *IEEE Transactions on Neural Networks and Learning Systems.* Under review. [A*; Q1]

Conference Papers

C-1. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Multi-source domain adaptation with distribution fusion and relationship extraction,” *in Proceedings of the International Joint Conference on Neural Networks (IJCNN).* Virtual online: IEEE, July 19 - 24 2020, DOI: 10.1109/IJCNN48605.2020.9207556. [A]

C-2. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Multi-source domain adaptation with fuzzy-rule based deep neural networks,” *in Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Virtual Online: IEEE, July 11 - 14 2021, DOI: 10.1109/FUZZ45933.2021.9494586. [A]

C-3. **Keqiuyin Li**, Jie Lu, Hua Zuo, and Guangquan Zhang, “Source-free multi-domain adaptation with generally auxiliary model training,” Proceedings of the International Joint Conference on Neural Networks (IJCNN). Padova, Italy, July 18 - 23 2022, DOI: 10.1109/IJCNN55064.2022.9892718. [A]

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Background

Machine learning becomes an effective and powerful tool for data mining especially in the situation where the original data is out of structure with high volume, high verity and high velocity. To make machine learning methods work well on learning tasks, there is always an assumption that the training data (source domain) and test data (target domain) have the same feature space or follow the same distribution. However, in practice, for a target task, sufficient labeled training data drawn from the same feature space or same distribution cannot always be found because of the high cost of collecting labeled data, or because that sometimes, the original data cannot be accessed considering privacy issues. Thus the traditional machine learning methods might lose their power. To solve this problem, transfer learning gains attention which can transfer knowledge from a source domain to another similar target domain, where source and target domains follow different distributions and have different original feature spaces or different label spaces.

One crucial condition to the success of domain adaptation is that the source and target domains can be connected closely. In consideration of this, three central issues of transfer learning proposed in previous research (Pan and Yang, 2009) play important roles for achieving domain adaptation: *when*, *what* and *how*. *When* to transfer identifies whether the source and target domains are related or if the knowledge from a source task will benefit the target task; *what* to transfer ascertains what kind of knowledge is appropriate for transferring across different domains; and *how* to transfer provides an algorithm to collect transferable knowledge and accomplish the transfer.

In relation to when to transfer, a theoretical study (Ben-David et al., 2010) analyses the precise state required for transferring a classifier between source and target domains and gives a bound of a classifier on the target domain error, taking account of its source domain error and the source-target distribution divergence. In (Gretton et al., 2012a), the researchers build a framework that measures the discrepancy between two distributions, which has been a prominent tool for comparing the data from the source and target domains to determine any similarity. Based on this, deep kernels (Liu et al., 2020a) parameterized by deep neural networks are explored to extend the measurement to fit data with a high dimension and complex structure. To explore the transferability of deep representations, an experiment-based study (Liu et al., 2019b) investigates the condition of transferring pre-trained networks via changing inputs and labels.

In deeming what to transfer, there are four types of transferable knowledge to consider: instance, feature, parameter and relationship (Pan and Yang, 2009). Instance-based methods focus on re-weighting source samples during training to guarantee the importance of those closer to the target domain. A represent study is boosting for transfer learning (Dai et al., 2007b), a boosting-based learning algorithm which utilizes some newly labeled data that follows the same distribution of target domain to leverage acknowledgement obtained from source domain and construct target task with high quality. Some recent instance selection methods, for example, transitive transfer learning and distant domain transfer learning (Tan et al., 2015, 2017), deal with transfer learning where source and target domains have little overlap, and connect them using auxiliary concepts.

Feature-based methods, the most widely explored category in transfer learning, aim to learn a robust representation of the source and target domains by transforming the original data into the same latent feature space (Wu et al., 2017). According to the original feature spaces of source and target data, it can be divided into homogeneous feature based methods and heterogeneous feature based methods (Weiss et al., 2016; Liu et al., 2018, 2020b). Homogeneous transfer learning means the dimensions of source and target features are the same, while that of heterogeneous transfer learning are different. A typical study is the multi-device indoor localization problem proposed in (Zheng et al., 2008), instead of assuming that the original data spaces of multiple tasks are similar, it hypothesizes that the latent feature spaces of related spaces can be similar. With development of deep learning (Krizhevsky et al., 2012; He et al., 2016), recent feature based transfer learning methods are combined with pre-trained deep networks to enhance the transferability of latent features (Long et al., 2015; Ganin and Lempitsky, 2015; Sun and Saenko, 2016).

Parameter-based methods primarily discover the shared parameters or prior distributions of two domains. A popular approach is joint active learning (Li et al., 2012), a support vector machine based method applying to cross-domain video concept detection, which combines the generative query strategy and traditional discriminative query strategy. Recent studies mostly employ parameter based method to solve transfer learning problem in the situation where the source access is unavailable (Lee et al., 2019).

Relationship-based procedures assume that some relationships between source and target domains are similar and employ available statistical relational learning techniques. Under this assumption, statistical relational learning techniques based on Markov logic become the dominating methods (Mihalkova and Mooney, 2008; Davis and Domingos, 2009). More recently, a framework has been built to identify the transferable knowledge in deep neural networks (Jang et al., 2019).

Regarding how to transfer, according to learning methods, it is known the algorithms contain neural networks (Kouw and Loog, 2019; Zhao et al., 2020b), Bayes and fuzzy systems (Lu et al., 2015; Wang et al.). Deep neural networks are widely explored, including convolutional (Wang et al., 2020b) and graph neural networks (Ma et al., 2019). Attention bridging based on convolutional neural network transfers knowledge from a single-labeled source domain to a complex multi-labeled target domain using visual attention mechanism, which can learn from not only image-level labels but also from the region-level information (Li et al., 2019b). Graph adaptive knowledge transfer model jointly optimizes target labels generated by the graph-based label propagation strategy and domain-invariant features collected using a semi-supervised class-wise adaptation strategy in a unified framework, which can benefit each other during training (Ding et al., 2018a). Lifelong learning (Wei et al., 2018), reinforcement learning (Keneshloo et al., 2019), adversarial learning (Dai et al., 2019) and meta-learning (Li and Hospedales, 2020) are applied to deep networks to enhance the transfer performance.

Transferring based on naive Bayes classifiers uses EM algorithm and Kullback-Leibler (KL) divergence to handle text classification task (Dai et al., 2007a), the former aims to estimate initial probabilities following the distribution of source domain, the latter aims to revise distribution of target domain using the learned probabilities. Transfer naive Bayes is a software defect prediction method which uses the knowledge of all the proper features in source domain to estimate the distribution of target data (Ma et al., 2012), then transfers learning information into weights of cross-company data and builds the corresponding prediction model. Model-agnostic meta-learning is treated as a hierarchical Bayesian model which can effectively adapt domains using the learned priors over task-specific parameters (Grant et al., 2018).

Fuzzy systems have superiority to deal with the ambiguity and permit the incorporation of approximation caused by the uncertainty oflearning tasks without precise information (Shell and Coupland, 2015). Since target labels are inaccessible, there is a limit to the amount of information with certainty that can be extracted, causing a high level of uncertainty in the target domain. Fuzzy multiple-source transfer learning deals with regression tasks in both homogeneous and heterogeneous scenarios with multiples source domains (Lu et al., 2020). It determines dominant source domains which contain more suitable transferable information for the given target domain by measuring the distance between each source and target class centres. Multi-source heterogeneous unsupervised domain adaptation extracts shared information from multi-dimension spaces using a novel shared-fuzzy-equivalence-relations neural network, and then transforms the acquired shared fuzzy knowledge into latent feature spaces to match the distribution discrepancy among heterogeneous domains (Liu et al., 2021).

In terms of the learning mechanism, there are three settings of transfer learning: inductive transfer learning, transductive transfer learning and unsupervised transfer learning (Pan and Yang, 2009). In the first case, source task and target task are different, no matter if the domains are different or not. With available labeled data in target domain, it still can be split into two categories: when the source labels are available, it can be regarded as multi-task learning (Zhang and Yang, 2017), which is aiming to leverage knowledge among multiple similar tasks and improve the performance of all tasks. While source labels are unavailable, it is similar to self-taught learning (Raina et al., 2007), which aims to employ unlabeled source data to improve performance on another labeled target data.

In the second case, the source task and target task are the same while domains are different. With unlabeled data in target domain and labeled data in source domain, the represent research category is domain adaptation (Zhang, 2019; Kouw, 2018), which aims to complete target task using knowledge learned based on source task by reducing discrepancy between source and target domains.

In the third case, source task and target task are different but related. With unavailable source data and target data, the main method is clustering (Dai et al., 2008), which aims to tackle unsupervised task in target domain with the assistance of unlabeled source domain.

Transfer learning attracts much attention and displays an upward tendency in a decade. However, there still are many research gaps. For instance, most existing studies focus on transfer learning with single source domain, but in practice, a target domain can be similar to multiple source domains which carry richer transfer information. In addition, for transfer learning with multiple source domains, the previous studies only explore similarities between source and target domains and complete target task by averaging source performance but fail to consider the differences and contributions of different source domains, which may harm the final performance and result in negative transfer. Third, most existing transfer learning methods assume source and target domains have the same label space, but how to transfer knowledge across domains where source and target domains contain unshared labels still needs further exploration, especially in non-satisfied and complex situation like data-free scenario. Finally, data privacy and security concerns resulting from sharing source data is ignored in most previous studies. When the source data is unavailable due to privacy issues, existing transfer learning methods relying on distribution matching cannot be applied. This thesis aims to tackle these problems by developing transfer learning methods with multiple source domains based on deep neural networks.

1.2 Research questions, objectives and expected outcomes

1.2.1 Research questions

To handle the mentioned problems for transfer learning and fill the research gaps in section 1.1, this thesis designs the following research questions (RQs):

- **RQ 1:** How to measure contributions of source domains to target domain?

  Source contribution is used to describe how important the source is when predicting the target task. It reflects the degree of similarity between source and target domains. The more similar the two domains are, the more contribution the source domain makes. This thesis focuses on transfer learning with multiple source domains, absolutely, the performance on target data of every source domain cannot be totally the same as each other because of the differences within source domains. Thus, if we desire to gain high-quality model for target domain, it is important to explore how to define contributions of different source domains, which means that the source domain performing superiorly on target domain should occupy the dominant position when predicting target task by combining all source performance.

- **RQ 2:** How to learn similarity of source and target domains to define what to transfer?

  Learning similarity, in other word, reducing discrepancy between source and target domains is the central idea of transfer learning, since the basic assumption for achieving transfer learning is that the source domain and target domain are related. However, without an appropriate similarity measurement, it may fail to match features or distributions of source and target data, which means the predictions of target data using model trained on source data show poor performance. Thus it is important to explore how to reduce discrepancy between two domains and guarantee the transfer performance. Measuring similarity between domains and selecting transfer knowledge can benefit to achieve transfer by reducing domain discrepancy.

- **RQ 3:** How to achieve transfer learning in heterogeneous setting?

  Many existing studies focus on transfer learning with homogeneous label space but will fail when applying to heterogeneous label settings directly. In practice, source domain which has the same label space with target domain cannot be always found. It is important to explore how to develop new approaches to solve transfer learning with heterogeneous label spaces.

- **RQ 4:** How to handle transfer learning when the source data is unavailable?

  Data security and privacy attract incredible attention in many areas and real-world applications. Traditional transfer learning relying on the access to source data to match distributions can trigger privacy concerns. Thus, it is necessary to explore how to transfer knowledge across domains without source data.

- **RQ 5:** How to explore soft information in transfer learning?

  Most existing transfer learning methods ignore the soft information resulting from uncertain data during transfer, which can be shared or can benefit the learning among multiple classes. As there is limited information available from target domain, data shift between source and target domains can cause a high level of uncertainty in the target domain, which can harm transfer performance. It is worthy to explore soft information to improve the positive transfer, which means learning in the source domain(s) facilitates learning in target domain.

1.2.2 Research objectives

To answer above research questions, this section sets up four research objectives (ROs):

- **RO 1:** Develop a set of frameworks to measure correlation and contributions of multiple source domains (to answer RQ 1).

  Many existing studies on transfer learning with multiple source domains complete target task by averaging performance of source domains. Although some weighted combination rules are employed, there are little disparity within source weights when combing their predictions, which means when the performance of source predictions have significant differences, the target predictor can fail to bring the superiority of source predictors that perform better on target domain into full play. Thus it is necessary to develop a set of frameworks to measure contributions of source domains and increase the disparity of weights to reduce negative transfer.

- **RO 2:** Develop a set of methods to select source samples which are more similar to the target domain by exploring the relationship of source and target domains. (to answer RQ 2).

  There are many techniques to measure discrepancy within two domains. However, when we apply these existing techniques to multi-source transfer learning, the learned similarities may disaccord from their performance on target domain, which means a source domain whose predictor works well on target domain may display little relatedness to target domain compared with other source domains with inferior predictors on target domain. Thus it is critical to improve the learning ability of similarity extracting and distill unrelated information during transfer to obtain high quality target predictor.

- **RO 3:** Develop a set of frameworks to transfer knowledge between source and target domains with heterogeneous label spaces (to answer RQ 3).

  Most previous studies focus on transfer learning with homogeneous label space, where source and target domains share the same label space. When comes to matching source model to the target domain with different label spaces, traditional methods developed based on homogeneous label space will fail because of the difference between dimensions of source and target labels. These encourage us to develop frameworks to handle transfer learning with heterogeneous label spaces to solve the mismatching problem.

- **RO 4:** Develop a set of frameworks to deal with source-free transfer learning (to answer RQ 4).

  Few previous studies focus on transferring knowledge across domains without source data. Existing previous transfer learning methods based on instance transfer and feature transfer cannot be applied entirely when there is no access to source data. To handle source-free transfer learning, we explore model adaptation methods based on parameter matching to transfer source knowledge to target domain.

- **RO 5:** Develop a set of frameworks to deal with soft information in transfer learning (to answer RQ 5).

  Existing transfer learning methods with and without source data rarely consider the data uncertainty in transfer learning caused by limited target information and data shift. Fuzzy system has the advantage of handling soft information. To eliminate the influence of data uncertainty, we explore a set of fuzzy rule-based transfer learning methods to enhance the transfer performance by exploring the soft information containing in features.

1.2.3 Expected outcomes

The desired outcomes of this thesis are as follows:

- Frameworks that could allocate weights of multiple source domains based on their performance on target domain.

- Methods to select transfer information by learning similarity of two domains.

- Frameworks that could handle transfer learning with homogeneous as well as heterogeneous label spaces.

- Approaches that could transfer knowledge across domains without referring source data.

- Frameworks that could handle soft information during transfer.

- A PhD thesis.

1.2.4 Research significance

The theoretical and practical significance of this thesis is summarized as following:

- Theoretical significance: This thesis develops approaches to measure the relationships between the source and target domains in transfer learning, which can be generally applied to existing methods. The proposed methods enrich the theoretical analysis of multi-source transfer learning, which explores the importance of both source samples and source domains, and proves the dominant role of the most similar source domain. Furthermore, it introduces sample and source distillation model. This work provides an idea to define what information is important to transfer in transfer learning, a problem has not been solved since transfer learning was proposed. Our research has implications for possible future work. Third, this thesis opens a new direction in transfer learning dealing with universal multi-source-free transfer learning, which can handle multiple source domains with both homogeneous and heterogeneous label spaces. It is the first work to solve transfer learning under universal multi-source-free setting where source domains have heterogeneous labels. Finally, this thesis introduces fuzzy model to deep neural networks to tackle source-free transfer learning, which enriches the theoretical and experiment analysis of fuzzy rules in knowledge transfer.

- Practical significance: The findings of this thesis contribute to real-world applications where there lacks enough labeled training data or exists data privacy concerns. All models developed in this thesis are validated on real-world datasets and tasks. Experiment results indicate the superiority of the proposed method compared with most existing methods. The source contribution measurement and sample and source distillation method can benefit transfer learning in applications with massive data, it helps select usable information and remove inefficient information to improve transfer learning performance while reducing the complexity of training resulting from large data size. The model based on fuzzy rules benefits real-world applications to make suitable decisions via considering all possible results.

1.3 Thesis Organization

The organisation of this thesis is listed in Fig. 1.1. Followed by detailed description.

- *Chapter 2*: This chapter presents a survey of transfer learning. Popular technologies of filling domain gaps in transfer learning are listed, including distribution matching based methods and parameter matching based methods. Then the categories of transfer learning are introduced in view of different standards. Methods of source-free transfer learning are reviewed at last.

- *Chapter 3*: Multi-source contribution learning is derived in this chapter. The proposed method deals with how to learn correlations and contributions of multiple source domains. Both common and diverse information from source and target domains are explored to adapt data on multiple distribution levels. A weight adjustment strategy and a fuzzy rule-based approach are developed to estimate the combination weights of combining source predictions when completing the target task. Experiments on real-world visual datasets are car-

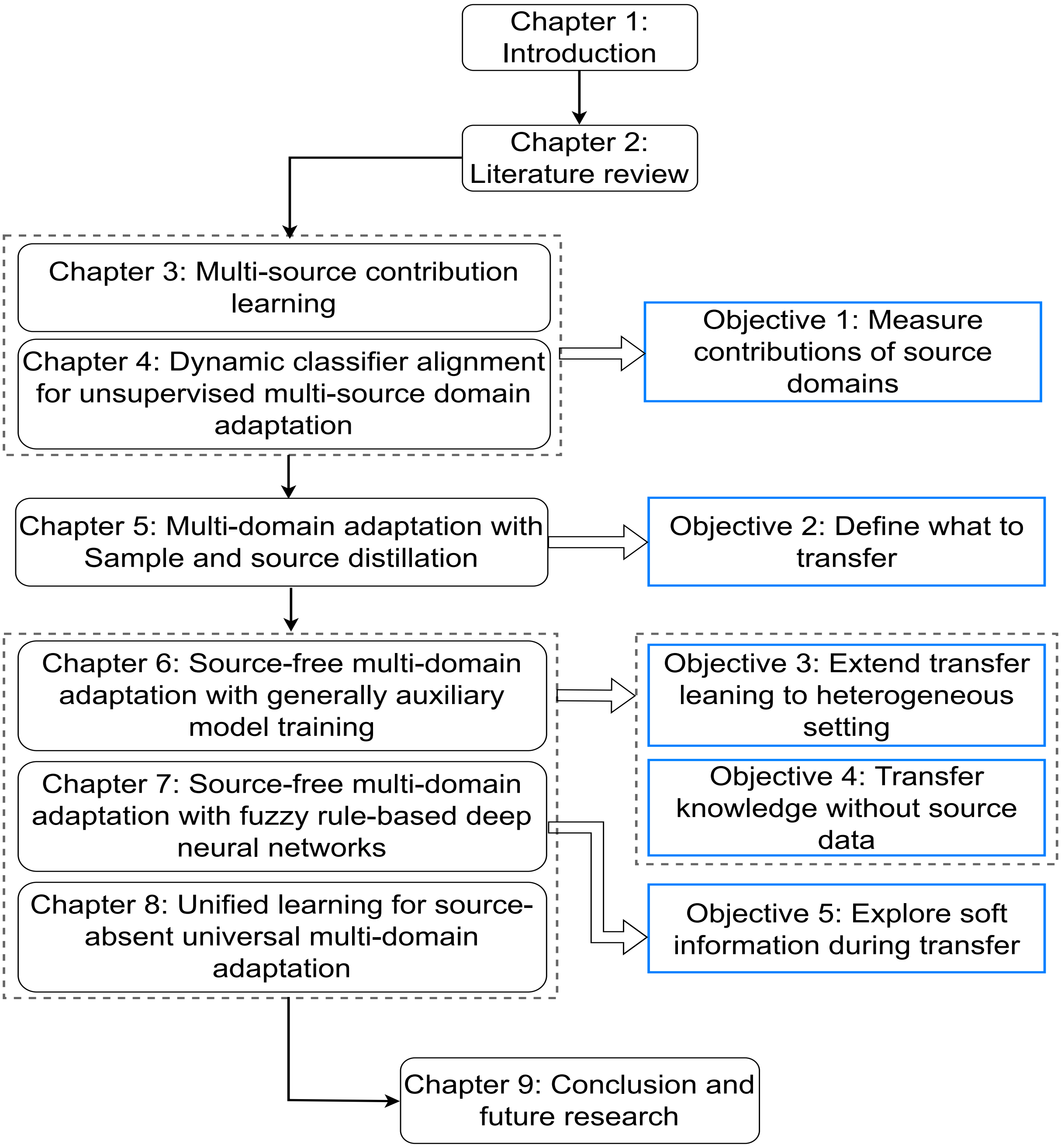

Chapter 1: Introduction
Chapter 2: Literature review
Chapter 3: Multi-source contribution learning
Chapter 4: Dynamic classifier alignment for unsupervised multi-source domain adaptation
Objective 1: Measure contributions of source domains
Chapter 5: Multi-domain adaptation with Sample and source distillation
Objective 2: Define what to transfer
Chapter 6: Source-free multi-domain adaptation with generally auxiliary model training
Chapter 7: Source-free multi-domain adaptation with fuzzy rule-based deep neural networks
Chapter 8: Unified learning for source-absent universal multi-domain adaptation
Objective 3: Extend transfer leaning to heterogeneous setting
Objective 4: Transfer knowledge without source data
Objective 5: Explore soft information during transfer
Chapter 9: Conclusion and future research

Figure 1.1 : Thesis organisation.

ried out to evaluate the proposed multi-source contribution learning method.

- *Chapter 4*: Dynamic classifier alignment is presented in this chapter. The proposed method deals with transfer learning with multi-view features which containing information extracted by different networks and multi-source domains, where both feature importance and source contribution are explored. The feature importance learning strategy is flexible to tackle multi-view features with the same or different dimensions. The dynamic classifier alignment builds a sample-wise method to learn source domain combination parameters. Experiments on real-world image classification tasks show the advantage of the proposed dynamic classifier alignment method.

- *Chapter 5*: Multi-source domain adaptation with sample and source distillation is developed in this chapter. The proposed method constructs a two-step selective strategy to eliminate negative transfer resulting from both source outlier samples and unrelated source domain(s). Simultaneously, the two-step selective strategy can identify the dominant source domain which is the most similar to the target domain. By defining the dominant source domain, the transfer knowledge from multiple source domains turns to the most similar knowledge from single source domain, which can reduce the parameter complexity in multi-source transfer learning and remove negative transfer from dissimilar domains at the same time. The proposed method is validated on real-world visual datasets and gains superior performance than most existing methods.

- *Chapter 6*: Multi-source-free domain adaptation with generally auxiliary model training under both homogeneous and heterogeneous label spaces is explored in this chapter. Generally auxiliary model is constructed from private specific source models to gain cross-domain ability, which is benefit to collect pseudo target labels with high confidence to self-supervise the learning of target model without the access to source data. To explore the influence of source sample quality, sample and source distillation is adopted to select similar samples and domains under source-free setting. Partial and open-set transfer learning with generally auxiliary model training are extended to deal with source-free transfer learning with heterogeneous label spaces. Experiments on real-world visual datasets indicate the superiority of the proposed generally auxiliary model training method under multiple transfer learning settings.

- *Chapter 7*: Multi-source-free domain adaptation with fuzzy rule-based deep neural networks under both homogeneous and heterogeneous label spaces is explored in this chapter. In the proposed method, source private model based on fuzzy rules of every source domain is learned by jointly training other source models using an auxiliary learning strategy, where source parameters are shared while source data is preserved. Furthermore, anchor-based alignment is designed to match target samples to the source anchors according to the agreements of clustering a target sample to a source category. Since source data is unavailable, to fit source models better, self-supervised learning based on pseudo labels is employed to train the target feature extractor which transforms target data into a latent feature space close to the source space. To reduce the influence of noisy target labels, a sample selection strategy is designed by combining the predictions of the source model and deep clustering to identify strong target samples, which are then used to update clustering centers that renew pseudo labels with a high level of certainty. Experiments on real-world visual datasets indicate the proposed fuzzy rule-based method is superior to non-fuzzy methods.

- *Chapter 8*: Unified learning for multi-source-free universal Domain adaptation is explored in this chapter. The proposed method deals with multiple source domains with homogeneous as well as heterogeneous label spaces. To classify known target samples without accessing to the source data, source generator is designed first to create source-like samples by combining global and local entropy assumptions based on contrastive learning, where local assumption aims to reducing the influence of unshared source categories. The generated source-like data is employed to match the target data under the supervision of target pseudo labels. To reduce pseudo label noise, the proposed method adopts center consistency and clustering to detect both source private and target unknown categories when pseudo-labeling the target data. By enlarging the distance between known and unknown samples, the performance on classifying known samples and detecting unknown categories can be guaranteed. Experiments on real-world datasets validate the superiority of the unified learning model.

- *Chapter 9*: A brief summary of the thesis and its contributions are given in the final chapter. Potential future studies are given as well.

Chapter 2

Literature Review

In this chapter, previous related works on transfer learning are briefly described. Section 2.1 introduces typical techniques for reducing data gaps in transfer learning, including distribution matching and parameter matching based methods. Based on the number of source and target domains, section 2.2 lists transfer learning methods dealing with single-source single-target, multi-source single-target and multi-source multi-target transfer learning. Homogeneous and heterogeneous transfer learning methods are introduced in section 2.3. Section 2.4 describes transfer learning methods tackle source and target domains with the same and different label spaces. In section 2.5, a new and challenging problem- source-free transfer learning- is presented. Following the settings of transfer learning, section 2.6 introduces commonly used learning schemes to achieve transfer across domains. Section 2.7 summarizes this chapter.

2.1 Discrepancy Measurement of Transfer Learning

To achieve transfer learning, the main idea is taking the advantage of the relatedness of two domains by reducing their discrepancy. Thus, discrepancy measuring becomes an essential operation of transfer learning. In this section, some discrepancy measuring techniques are introduced, including distribution matching based technology and parameter matching based technology.

2.1.1 Distribution Matching Based Methods

Distribution matching based methods are commonly developed based on maximum mean discrepancy (MMD) (Gretton et al., 2006), Wasserstein distance (Redko et al., 2019a; Arjovsky et al., 2017; Dai et al., 2019; Zhao et al., 2020a), Kullback-Leibler (KL) divergence (Pan et al., 2020) and H-divergence (Zhao et al., 2018; Wen et al., 2020). MMD based distribution matching is the most widely explored than the others (Ganin and Lempitsky, 2015; Lifshitz and Wolf, 2021). Distribution matching can be divided into single-level matching and multi-level matching. We first introduce single-level matching methods.

Single-level distribution matching mainly indicates adapting domains on domain level only. MMD is a typical method to test if two samples are drawn from the same distribution (Gretton et al., 2012a). Transfer component analysis first applies MMD to domain adaptation in order to achieve marginal distribution matching in a reproducing kernel Hilbert space (Pan et al., 2010), and gives a solution of MMD instead of using optimization solver. Deep domain confusion uses a pre-trained convolutional neural network to optimize the classifier and domain invariant features automatically by incorporating MMD into deep networks (Tzeng et al., 2014). Based on these, joint distribution adaptation and joint adaptation networks extend transfer component analysis to matching both marginal and conditional distributions (Long et al., 2013, 2017). Deep adaptation networks employ multi-kernel MMD (Gretton et al., 2012b) to reduce the domain bias and adapt all hidden representations of task specific layers to match the mean embeddings (Long et al., 2015). Improved MMD techniques such as central moment discrepancy (Zellinger et al., 2017), deep-kernel based MMD (Liu et al., 2020a) are extended recently to improve the ability of discrepancy measuring, the former mainly reduces the computational complexity, while the latter is more flexible for complex and high-dimension data. Adversarial learning adapts distributions by making two domains indistinguishable, which aims to obtain more stable and robust gradients in the situation where the distributions of source and target domains have no overlaps (Pei et al., 2018; Yu et al., 2019a). The graph-matching metric is developed as the domain discrepancy measurement which has the ability to map both nodes and edges between source and target representations. By doing this, not only is distribution knowledge considered, but also structural and geometric information which is rarely investigated in most previous studies is considered (Das and Lee, 2018; Yang and Yuen, 2019).

To reduce domain shift and improve the ability of source predictor on target domain, except for domain-level distribution matching, the adaptation on other distributions is proposed (Kang et al., 2019). Dynamic adversarial adaptation network learns different contributions of marginal and conditional distributions between domains dynamically, the former is achieved by training a global domain discriminator, while the latter is built by training several class-wise domain discriminators (Yu et al., 2019a). Multi-adversarial domain adaptation enables fine-grained alignment by training multiple domain discriminators, which forms multimode structures based on different data distributions of categories (Pei et al., 2018). Transferable attention network diminishes multiple region-level and single image-level distribution discrepancies, where multi-adversarial domain adaptation matches domain-level and class-level distributions by multi-mode discriminators (Wang et al., 2019b). Local feature pattern method jointly maps holistic feature distribution and local pattern distributions. These multi-level distribution matching technologies enable fine-grained alignment of cross-domain adaptation (Wen et al., 2019). Dual adversarial domain adaptation proposes a 2K-dimension discriminator which aligns both domain-level and class-level distributions simultaneously, and develops a mechanism to handle samples without discriminative features using multi-view learning and adversarial learning (Du et al., 2020). Pixel-level and feature-level adaptations are considered in domain mixup networks (Xu et al., 2020a). In pixel-level matching, each pair of source and target samples is linearly interpolated, and the mixed inputs with soft labels are used to train the domain discriminator via adversarial learning. In feature-level matching, the source and target embeddings are mixed to ensure domain alignment and category consistency simultaneously. Classification and clustering distribution adaptations are explored based on self-ensembling (Pan et al., 2020). It first divides target samples into multiple groups, then a source only model is performed on the target domain to provide both category and cluster assignment information. By matching the source global and local mutual information to the target simultaneously, a target classifier can be learned based on the source model.

2.1.2 Parameter Matching Based Methods

Distribution matching based methods are built on the assumption that transfer learning can be succeeded via matching distributions of two domains in the transformed latent feature spaces. However, as proven in the previous study (Ben-David et al., 2010), the model works well on both source and target domains may not exist even their distributions are matched. Considering adapting domains by matching distributions can suffer risk because of the non-discriminative features, asymmetric tri-training trains multiple classifiers using labeled data from source domain and then generates and updates artificial labels of unlabeled samples, the pseudo-labeled target domain is employed to predict target data using a independent network (Saito et al., 2017). Multi-source sentiment generative adversarial network forms a cycle-reconstruction pipeline using consistent adversarial learning and trains target model directly by minimizing the distance between the generated and true samples in latent space (Lin et al., 2020).

To handle the transfer situation where there is no access to original source data but only the pre-train model, fine-tuning based methods (Li et al., 2018c; Lee et al., 2019; Chin et al., 2020) are developed to minimize the loss of model parameters between source and target networks. Source hypothesis transfer develops a self-supervised labeling method to map feature learning of source and target data by freezing the prediction layer trained on source domain (Liang et al., 2020). Noisy feature distillation improves the ability of transfer model to deal with adversarial attacks caused by traditional fine-tuning methods, which generates clean data via random initialization during training (Chin et al., 2020).

2.2 Categories of Transfer Learning Based on Number of Domains

There are four types for transfer learning based on the numbers of source and target domains, including single source and single target (Zhao et al., 2014; Liang et al., 2019; Wang et al., 2022; Xiao and Zhang, 2021), single source and multi-target (Yu et al., 2018; Tian et al., 2020b), multi-source and single target (Lu et al., 2020; Zhao et al., 2019b; Zhou et al., 2021; Xu et al., 2020a; Feng et al., 2020), and multi-source and multi-target transfer learning. Since there are very few studies focusing on multi-source and multi-target transfer learning, here we introduce the former three settings, respectively.

2.2.1 Single Source and Single Target Transfer Learning

Single source and single target transfer learning (Chen et al., 2020; Li et al., 2021d) has progressed considerably for varying learning tasks such as classification (Wen et al., 2019), segmentation (Sun et al., 2019) and regression (Zuo et al., 2016). Based on feature and parameter transformation, one popular technique to achieve transfer learning is domain adaptation (Ben-David et al., 2007; Kouw, 2018; Zhang, 2019).

A novel metric function named central moment discrepancy is proposed to measure the distance between probability distributions which can be solved without highly complex and costly kernel matrix computations (Zellinger et al., 2017). Fuzzy system combined with granular computing is used to achieve regression transfer in homogeneous and heterogeneous feature spaces (Zuo et al., 2017, 2018b). Benefiting from the development of deep learning, recent surveys employed deep neural networks to extract common features of source and target domains. To explore the efficiency of deep structures, transferability of deep feature representations is explored. Experiments as well as theory analysis on what and where to transfer in deep networks are provided in recent studies (Liu et al., 2019b; Jang et al., 2019). Multi-representation adaptation network aligns the distributions of source and target representations by a multi-structure network which extracts representations from different aspects (Zhu et al., 2019b). Joint geometrical and statistical alignment is presented to unify shared space and subspaces of source and target domains via reducing the shifts of the geometries as well as the distributions simultaneously (Zhang et al., 2017). Dynamic distribution adaptation develops a strategy to evaluate the importance of the cross-domain marginal and conditional distributions (Wang et al., 2020b). This work explores both traditional and deep transfer learning. Traditional transfer learning based on manifold space is developed by extracting Grassmann manifold features which contain more details and property of domains. Geodesic flow kernel is used to reduce computational complexity when transforming data into manifold space. Joint distribution weights are calculated based on geometrical property controlled by Laplacian regularization. Deep dynamic distribution adaptation network constructs an end-to-end structure that leverages the ability of both feature extractor and classifier. Alignment weights of multi-level distributions are estimated based on A-distance.

2.2.2 Multi-Source and Single Target Transfer Learning

Multi-source domain adaptation involves more difficulties and challenges than single source domain adaptation because simply combining source samples with domain shifts is inappropriate without knowing the mixture parameters of the target distribution (Mansour et al., 2009). For multi-source domain adaptation, combining all source domains as one and treating it as single source domain adaptation is a simple way. However, this operation fails to consider the domain shift within source domains, which can result in negative transfer (Redko et al., 2019b). In order to solve the mentioned problem caused by domain shift and achieve performance of target predictor which is desired to be higher than using single domain only, multi-domain matching network designs a domain adaptor which matches distributions within all domains by learning their relationships and turns the learned information into weights of each two domains, the final transfer knowledge is extracted from a subset of source domains which gain larger weights (Li et al., 2018d). Moment matching for multi-source defines moment distance to adapt two domains, which not only considers relatedness between source and target domains, but also among source domains using adversarial learning, and creates a new multi-domain dataset to verify the proposed method (Peng et al., 2019a). Multiple feature spaces adaptation network aligns cross-domain distributions and designs cross-domain classifier constraints to improve the performance on target samples which are close to class boundaries (Zhu et al., 2019a).

Many multi-source transfer learning methods complete target task by averaging the predictions of source domains. However, different sources usually perform differently from each other when applying to target domain. Thus, it is reasonable to define the weights according to their contributions to target domain. Multi-source selective transfer method employs three re-weighting strategies to choose appropriate source domains and combine them to complete target task, including nearest selection, weighted selection and Top-k selection (Zhang et al., 2019a). Deep cocktail network proposes a new setting named category shift, which means the label space of each source domain is a subset of the label space of target domain, the union of source label spaces covers the target label space (Xu et al., 2018). Under this assumption, it uses source-target-specific perplexity scores to re-weight source distributions and represents target distribution by weighted mean combination rule. Multi-source distilling domain adaptation distills source samples by measuring the similarity between source data and target domain, only closer samples are used to fine-tune the pre-trained model and estimate target labels by re-weighting sources using standard Gaussian Distribution based on the learned distance between each source sample and target domain (Zhao et al., 2020a). Multi-source adversarial domain aggregation network trains one model for all domains by combining all adapted domains together closely using adversarial domain aggregation, which avoids learning weight of each source domain and optimizes the model by minimizing losses of the two discriminators (Zhao et al., 2021).

To take advantage of every source domain, multi-source transfer across domains and tasks proposes the gradient mixing strategy based on meta-learning (Li et al., 2020a). Instead of adding extra constraints for distribution matching, it weights and mixes the gradients from all the source domains using an online method to preserve transferable knowledge during training. Hard and soft labeling approaches are employed to collect pseudo labels predicted by multiple pre-trained models. Domain generalization with adversarial feature learning extracts domain-invariant representations from multiple source domains to learn a universal classifier to be performed on an unseen target domain, where there is entirely no target data available during training (Li et al., 2018a). Mixture of multiple latent domains for domain generalization is developed to tackle a novel case where the label of a sample belonging to a domain is unknown, and adversarial learning is employed to extract shared features from pseudo labeled domains divided by clustering (Matsuura and Harada, 2020) . Multi-source domain adaptation with graph embedding and adaptive label prediction measures the inter-domain discrepancy on the feature-level and intra-domain discrepancy on sample-level simultaneously by jointly optimizing moment matching and geometry alignment. K-means clustering and the nearest neighbor classifier are used to predict the pseudo target labels in multiple feature spaces, which are later taken as inputs to update the model (Ma et al., 2020). Dynamic transfer is proposed to deal with multi-source domain conflicts resulting from a domain-agnostic model by adapting the model parameters to samples rather than across domains, where the alignment between source and target domains can be simplified by turning multi-source domains into a single-source domain (Li et al., 2021e). Multi-source domain adaptation with guarantees builds a global teacher model by combing local source experts, and reduces the domain gaps using an adversarial learning method where a student model is trained to mimic the teacher expert (Nguyen et al., 2021).

2.2.3 Single Source and Multi-Target Transfer Learning

The main challenge of multi-target transfer learning is the transfer knowledge from one source domain may be inadequate to multiple target domains. To solve this problem, complementary knowledge is adopted to build target common model to assist in fitting target individual models, where parameter adaptation framework is developed by introducing sparse dictionaries (Yu et al., 2018). The parameter adaptation framework first learns bridging parameter dictionary between each pair of source and target domains, then target common dictionary is generated by minimizing the distance among target common and individual dictionaries. By this, the knowledge from labeled source and unlabeled target domains can be extracted simultaneously. Heterogeneous graph attention network deals with semantic association among unlabeled target domains ignored by pairwise adaptation methods using a deep semantic information propagation approach, which takes advantages of attention mechanism to improve the transductive ability of graph network and optimize the semantic transfer among source and multiple target domains. The unified target subspace is constructed to predict pseudo target labels, and then the domain invariant information extracted based on pseudo labels is employed to align semantic knowledge from source and target domains (Yang et al., 2020b).

2.3 Categories of Transfer Learning Based on Feature Space

According to the dimensions of the feature spaces of source and target domains, the feature-based approach can be divided into homogeneous and heterogeneous domain adaptation (Xu et al., 2022; Liang et al., 2021a; Liu et al., 2020b).

2.3.1 Transfer Learning with Homogeneous Feature Space

In homogeneous transfer learning, the source and target feature spaces have the same dimension. Structurally regularized deep clustering proposes a source regularized method for unsupervised domain adaption (Tang et al., 2019). Motivated by the structural similarity, it employs a deep clustering framework to learn class centres of source and target domains, and generates an auxiliary target distribution to help explore the intrinsic discrimination in the target domain by matching it to the source distribution. Certainty-based attention for domain adaptation identifies adaptable regions by building a Bayesian discriminator (Kurmi et al., 2019). The predominant areas which can benefit the matching of source and target data are highlighted by the class probabilities returned using a Bayesian classifier. Dynamic weighted learning introduces a degree of alignment and discriminability to avoid the discriminability vanishing problem, and adopts sample weights to deal with sample imbalance across domains (Xiao and Zhang, 2021). Faster domain adaptation aims to reduce computational cost in transfer learning, which improves the efficiency of the energy-sensitive platforms from two aspects, including optimizing domain transfer network and the layer selection function, where early stopping and amid skipping are employed to decrease the time and energy costs (Li et al., 2021b).

2.3.2 Transfer Learning with Heterogeneous Feature Space

Heterogeneous domain adaptation means features spaces of source and target domains have different dimensions, both offline (Wei et al., 2016; Luo et al., 2017) and online heterogeneous transfer learning are explored (Wu et al., 2017; Yan et al., 2017). Completely heterogeneous transfer learning deals with domain adaptation where both the feature and label spaces of the source and target domains are different (Moon and Carbonell, 2017). Transfer independently together gives large weights of pivot samples and low weights of outliers based on graph optimization, which designs a projection matrix to solve the mismatching problem of source and target data (Li et al., 2018b). Deep matrix completion with adversarial kernel embedding employs an auto-encoder structure to map features in latent space by an adversarial kernel and handles the mismatching problem by a matrix completion way to reconstruct the missing values (Li et al., 2019a). Generalized deep transfer networks dealing with heterogeneous domain adaptation, which transfer the acquired label knowledge from textual domain to visual domain, are proposed by introducing hidden shared layers based on parameter and representation, respectively (Shu et al., 2015; Tang et al., 2016). Various variants with different fine-tuning schemes are explored to improve the ability of the proposed structures when handling image-text pairs, and a new dataset is created to validate the method. Discriminative distribution alignment derives a domain invariant space to match domain discriminative directions as well as distributions (Yao et al., 2020). To separate class samples, an adaptive classifier is trained by reducing conditional distribution divergence and enlarging distance between class centers. Both cross-entropy and squared loss are employed to parameterize the training. Structure and classification space alignment adapts domains on both data-level and parameter-level, which preforms feature space matching, distribution alignment and classifier alignment jointly in a unified framework (Tian et al., 2021). Target-oriented classifier is generated from supervised source domain and semi-supervised target domain using a balance factor, which gives weight to the target domain in view of its similarity to the source domain during adaption. Label distribution alignment is introduced using respective projection matrices to avoid feature dimension heterogeneity. Spatial structure preservation is employed to enlarge distance among samples from different classes.

2.4 Categories of Transfer Learning Based on Label Space

In view of label spaces of source and target domains, transfer learning can be divided into closed set (Rozantsev et al., 2018), open-set (Saito et al., 2018), partial (Li et al., 2021d) and universal (You et al., 2019) transfer learning. In closed set transfer learning, source and target domains have the same label spaces. In open-set transfer learning, source label space is a improper subset of target label space. In partial transfer learning, source label space contains target label space. In universal transfer learning, both source and target domains have their own private label spaces and share the intersection of label sets. In this thesis, closed set is also named as homogeneous label space setting, while open-set, partial set and universal set are grouped as heterogeneous label space settings Azizzadenesheli et al. (2019); Sohn et al. (2019).

Complement label and incomplete label spaces are special settings of heterogeneous label spaces. Complement label generated by adversarial network is adopted to handle transfer learning where acquiring the fully true labels of the source domain is overpriced, the generated complementary-labeled source data is used to replace unavailable fully-true-labeled data (Zhang et al., 2020). Incomplete label is the open-set setting with multiple source domains. To solve incomplete label space among multiple source domains, low-rank matrix based on latent features from target domain is employed to recover missing labels in source domains (Ding et al., 2016).

In the following, transfer learning under closed set, open-set, partial and universal scenarios are listed.

2.4.1 Closed Set Transfer Learning

Closed set transfer learning is the most widely explored. Pseudo labelling strategy becomes popular in transfer learning recently. Transferable prototypical network collects pseudo labels of the target domain by finding the nearest prototype from source domain for each target sample, and minimizes the distance between the prototypes for each category in source and target domains as well as their distribution scores to adapt domains (Pan et al., 2019). Cluster alignment with a teacher matches both marginal distribution and class-conditional structure of the source domain to that of the target domain, which enhances existing unsupervised domain adaptation methods by aligning the clusters across the source and target domains with the help of the pseudo labels provided by the teacher model (Deng et al., 2019). Enhanced transport distance, which parameterizes the Kantorovich potential value, has been developed to measure domain discrepancy. The attention mechanism is used to reweight the distance matrix according to the degree of correlation between samples (Li et al., 2020c). Domain adaptation with gradually vanishing bridge introduces bridge layers and intermediate domain during adversarial training (Cui et al., 2020). It is expected that the intermediate domain which provides invariant information can cover almost all source and target samples after minimizing domain boundaries using the bridge. Adversarial-learned loss for domain adaptation attempts to fill the gap between the pseudo and ground truth labels by introducing a confusion matrix (Chen et al., 2020). Simultaneously, a regularization term classifying the source samples is added to the discriminator learning to improve robustness. Progress domain adaptation employs a self-learning method to progressively update target model constructed from pre-trained source model, which defines target class prototypes by selecting reliable samples with lower self-entropy, where set-to-set distance-based filtering is adopted to reduce the noise of pseudo target labels (Kim et al., 2020). Instance level affinity-based transfer discovers similar and dissimilar samples between the source and target domains, where K-nearest neighbor-ranking is used to pseudo-label target samples and build the affinity matrix, which forces intra-class grouping and inter-class separating (Sharma et al., 2021)

2.4.2 Open-Set Transfer Learning

Open-set transfer learning or zero-shot transfer learning, aims to recognize target categories containing shared labels which are the same as that of source domain and unknown labels which never show up in source domain (Panareda Busto and Gall, 2017; Busto et al., 2018; Kundu et al., 2020b; Bucci et al., 2022). Separate to adapt solves open-set domain adaptation by taking the openness of the target domain into account, which develops a coarse-to-fine weighting separation mechanism to recognize unknown samples from known samples by learning similarities between target data and each source category (Liu et al., 2019a). Data with high similarity is regarded as known category, while data with low similarity is regarded as unknown category. Distribution alignment with open difference achieves open-set domain adaptation using structural risk minimization principle and open set difference regularization which estimates the generalization bounds controlled by maximum mean discrepancy based on theoretical analysis (Fang et al., 2021). Mutual to separate employs a dual-control system to select unknown categories from the known ones, in which the sample separation network filters out unknown samples while the distribution matching network maximizes the domain confusion (Chang et al., 2020). Domain-augmented meta-learning framework deals with open domain generalization problem, where knowledge extracted from distinct source domains is transferred to an unseen target domain (Shu et al., 2021). To fill domain gaps, meta-learning is adopted to minimize the distance of generalizable representations rather than distribution distance among source and target domains. Feature-level and label-level augmentations are developed to overcome the disparate label sets caused by minor class in open domain generalization. Novel target discovery method explores the underlying structures from seen classes and interpretable semantic attributes from unseen classes simultaneously, where partial alignment is preserved to mitigate domain shift when target label space is larger than source label space. Attribute propagation is proposed to discover visual semantic matching for unknown classes based on graph structure (Jing et al., 2021). Soft unknown-class rejection method overcomes the sensitivity caused by predicting unknown classes according to crucial hyperparameters, which assigns soft weights to target samples in view of their entropy values (Xu et al., 2021b). Progressive graph learning deals with open-set transfer learning without source data, which decomposes shared and unknown subspaces of target domain to reduce source partial risk and progressively reject target samples with low confidence as unknown classes to eliminate open-set risk. At the same time, both graph-structured sample-level and manifold-level distributions are aligned to fill conditional shift between domains (Luo et al., 2020, 2022).

2.4.3 Partial Transfer Learning

The challenge of partial transfer learning is how to identify samples from shared categories to transfer knowledge across domains and remove samples from unshared categories to reduce negative transfer. Selective adversarial network minimizes the discrepancy between the source and target distributions in the shared label space and filters out unshared labels using the multi-discriminator domain adversarial network (Cao et al., 2018a). Graph partial domain adaptation network develops a label relational graph controlled by the moving average centroid separation constraint to match the feature distributions as well as the data structure of the shared categories (Yang et al., 2020a). Example transfer network proposes a transferability weighting framework to discover shared label space automatically in terms of the similarity between each source sample and the target domain (Cao et al., 2019). Domain adversarial reinforcement learning gradually selects source shared samples by introducing deep Q-learning strategy controlled by the action-value function, which designs a reward framework to guide the selection policy to automatically select usable source samples to adapt to the target domain (Chen et al., 2022b). Multiple self-attention network extracts both effective high-level context features and low-level structural features by introducing a gradual feature enhancement manner based on self-attention module to adversarial learning, which filters out unshared source samples using multiple domain discriminators with a weighting scheme (Zhang and Zhao, 2021).

2.4.4 Universal Transfer Learning

Universal transfer learning is a pretty new setting, the main challenge is how to divide shared categories from private categories of each domain. The early research on universal transfer learning is universal domain adaptation, which develops an end-to-end solution based on a weighting mechanism by exploring both similarity and prediction uncertainty to discover shared label set (You et al., 2019). Universal multi-source adaptation network designs a novel pseudo-margin vector to select reliable samples belonging to shared label set, which aligns multiple source and target domains via adversarial learning, and a theory analysis on loss function is provided (Yin et al., 2022). Domain consensus clustering divides shared categories and private categories by exploiting both semantic-level and sample-level consensus knowledge, where cycle-consistent clusters and cross-domain classification agreement are used to determine common classes and private classes, respectively (Li et al., 2021a). Domain adaptative neighborhood clustering via entropy optimization combines neighborhood clustering with entropy-based feature alignment and rejection to select common class samples and reject unknown samples, where self-supervised learning is adopted based on the pseudo labels provided by neighborhood clustering to match each target sample to source domain (Saito et al., 2020). Universal source-free domain adaptation develops a two-stage learning process to solve universal domain adaptation without source data, where the procurement stage leverages available source data to enhance the rejection of out-of-source distribution samples by building a generative classifier framework, while the deployment stage operates wide range of category-gaps by defining an instance-level weighting mechanism to adapt domains (Kundu et al., 2020a). One-vs-all network learns a closed-set classifier to categorize known classes which is parameterized by cross-entropy loss, and a classifier for each source class to define the boundary between the positive and the nearest negative samples by minimizing open-set entropy which determines a threshold based on the assumptions of all classifiers to reject unknown classes (Saito and Saenko, 2021). Active universal domain adaptation not only categorizes known classes but also recognizes the unknown classes, where adversarial and diverse curriculum learning are used to train source model which predicts known samples. To infer target-private labels, a small budget of annotated target samples provided by active learning is adopted to assist in dividing unknown classes (Ma et al., 2021b). Universal model adaptation learns a two-head classifier from the source domain and applies it to the target domain with an informative consistency score to divide known and unknown samples in the target domain Liang et al. (2021b). In the source model training procedure, a closed-set classifier is learned to predict the softmax class probability, while in the model adaptation process, a threshold defined by the mean informative consistency is used to select unknown samples.

2.5 Source-Free Transfer Learning

To handle data privacy concerns, source-free domain adaptation is proposed Liang et al. (2020); Zhao et al. (2019a); Liang et al. (2021b, 2022); Wu et al. (2021); Ahmed et al. (2022). Two approaches are commonly employed to transfer knowledge across domains without source data, including data generation Li et al. (2020d) and model adaptation Yang et al. (2021c). Data generation methods are developed based on generative adversarial network Hou and Zheng (2020). The central idea is constructing source-like or target-like samples using a generator, then reducing the difference between the generated and real samples by a discriminator to adapt the source model to the target domain. A recent study- source data free domain adaptation- learns joint distribution of source domain by producing source-style proxy samples from the pre-trained source classifier. The learned distribution is then used to extract invariant features of the unlabeled target domain to fine-tune the pre-trained model Kurmi et al. (2021a). Model adaptation methods depend mainly on a pseudo-labeling strategy Wang et al. (2022). Unsupervised learning techniques, such as clustering, are employed to provide pseudo target labels. The target model is then trained based on the source model in a self-supervised way. A previous multi-source data free domain adaptation method adopts weighted information maximization and weighted pseudo-labeling to combine source predictions automatically and collect target labels, the target model is trained by jointly optimizing the source feature encoders with corresponding weights Ahmed et al. (2021).

Commonly used pseudo-labeling techniques includes clustering and K-nearest neighbors. To perform source model on a target domain without source data, robust adaptation is proposed to preserve the robustness and performance of the pre-trained source model, where standard models are employed to provide pseudo target labels with less noise, while robust models are used to generate adversarial target samples which expect to enhance the domain alignment Agarwal et al. (2021). Casting a BAIT deals with both online and offline source-free domain adaptation by building a two-step optimization policy, where an extra classifier which identifies certain and uncertain features is introduced to find misalignment samples, while the multi-class source classifier is used to provide class anchors Yang et al. (2021b). Universal black-box domain adaptation converts the target task into sub-tasks, including in-class discrimination and out-task detection, where the in-class task learns a multi-class classifier to categorize target labels in known classes, while out-class task learns a binary classifier to reject unknown classes (Deng et al., 2021).

2.6 Methods for Transfer Learning

2.6.1 Weakly-Supervised Learning Methods

Weakly-supervised learning is one of the powerful algorithms to handle transfer learning in the situation where the domain has a proportion of weighted inaccurately labeled or incomplete labeled samples by training multiple weak predictors (Zhang, 2019).

Semi-supervised transfer learning is a typical method of weakly-supervised learning (Kipf and Welling, 2016; Yao et al., 2015; Xiao and Guo, 2014). Instance constraints, an adaptive SVM based transfer learning method including projective model transfer SVM and max-margin domain transforms, is proposed to transfer classifiers learned from a source domain containing available labeled and unlabeled samples (Donahue et al., 2013). Generalized distillation semi-supervised domain adaptation transfers knowledge obtained from the unlabeled data to target task without the access to source domain (Ao et al., 2017). Soft labels of target data and SVM which is treated as the base classifier are employed to solve the problem and estimate parameters. Minimax entropy approach is proposed to solve contemporary domain adaptation where there are very few labeled samples in source domain, which estimates domain-invariant prototypes by minimizing the discrepancy between the class prototypes and the unlabeled target domain (Saito et al., 2019).

Active learning is another attractive weakly-supervised learning method that leverages knowledge gained from the selected informative unlabeled samples from a target domain (Settles, 2009; Yang et al., 2013; Wang et al., 2014), which generatively labels the chosen samples and uses them to form a new training set combining with the existing labeled samples. Multi-kernel learning with active learning develops a hyperspectral image classification method based on active learning and domain adaptation, which aims to learn a multi-kernel classifier using newly formed training data containing labeled source samples and selected user-labeled target samples (Deng et al., 2018a). Deep joint spectral-spatial feature learning handles image classification task using hierarchical stacked sparse auto-encoder networks and active learning. The former aims to extract specific discriminative features, the latter aims to transfer the learned information based on limited labeled samples from both source and target domains (Deng et al., 2018b). Infinite Gaussian mixture model with active learning takes advantage of Gibbs sampling strategy and the interactive query strategy to identify the data correlation in source and target domains and enhance the transfer performance (Zuo et al., 2018a). Heterogeneous transfer learning through active correspondences construction handles cross-language text classification by exploiting correspondences between source and target domains to complete low-rank matrices and reconstruct the generate unprecise target samples (Zhou et al., 2016).

2.6.2 Unsupervised Learning Methods

This section lists two kinds of unsupervised transfer learning methods, including deep learning based methods and fuzzy system based methods.

Deep learning has been used in many applications, and now becomes the main trend of most research fields related to artificial intelligence (LeCun et al., 2015; Zhuang et al., 2020; Caron et al., 2020), such as nature language processing (Dong et al., 2019; Malte and Ratadiya, 2019), image processing (Qin et al., 2019; Jing and Tian, 2020), recommendation system (Hu et al., 2018a), biological application (Gupta et al., 2020; Yu et al., 2019c) and so on. To explore the transferability of deep networks and find when and where to transfer, previous studies have provided some related theory analysis and experimental verifications (Liu et al., 2019b; Jang et al., 2019). Here we divide deep learning based transfer learning methods into three types: convolutional neural network based method, generative adversarial neural network based method and graph neural network based method.

For convolutional neural network based methods, pre-trained deep networks are widely employed to extract shared feature of source and target domains (Zhang et al., 2017; Zhu et al., 2019b; Kurmi et al., 2019). Transfer channel pruning aims to reduce the high computational cost of deep transfer networks by compressing the model, which removes the less important parameters of shared network by re-weighting contribution degree of channels (Yu et al., 2019b).

For generative adversarial neural network based methods, a typical study is generative domain adaptation network (Gong et al., 2018). It captures distribution changes between two domains and generates new data for the extracted latent features. To avoid high dimension problems caused by generating all features together, the improved method causal G-DAN is developed to collect low dimension data by decomposing the joint distributions into separate modules. Deep adversarial attention alignment network combines attention alignment and cycle generative adversarial network to transfer knowledge in all hidden layers under the assumption that discriminative information in an image rarely changes with the style of the image (Kang et al., 2018). Generative pseudo-label refinement exploits the ability of conditional generative adversarial network and applies it to transfer learning to deal with noise of pseudo labels of target domain resulted from domain shift when using model trained on source domain, which can generate clean target samples by adversarial learning (Morerio et al., 2020). Domain impression builds a generative framework to deal with source-free domain adaptation with noise (Kurmi et al., 2021b). It includes both generation and adaptation modules. The generation module first obtains samples that can be divided correctly by the source classifier to train a discriminator. Then the adaptation module fits the source classifier to the target domain by minimizing the likelihood loss using an adversarial way.

Graph neural network is a new type of deep network that differs from convolutional neural network (Kipf and Welling, 2016; Wang et al., 2019c; Ding et al., 2018a; Das and Lee, 2018; Ma et al., 2019), transfer learning based on this structure still need further exploration. Adversarial domain adaptation with graph convolution handles node classification by combining adversarial learning and graph convolution, which extracts network invariant representations by minimizing the Wasserstein-1 distance between source and target domains instead of the binary classification in original adversarial networks (Dai et al., 2019). By doing this, it is desired to improve the stability of the model. Domain adaptation through graph method constructs auxiliary domains based on graphs to describe the dependencies among domains, and introduces metadata information to deep network structures to explore the relation between source and target samples (Mancini et al., 2019). Learning to combine explores the interaction among domains by constructing graph-structured data, where prototypes of multiple domains are combined to discover the information propagation (Wang et al., 2020a). Self-supervised graph neural network builds a multi-source domain adaptation structure by connecting the self-supervised and the target tasks, which employs mask token strategy to take each sourced domain as a token, and predicts a domain based on random masking domain information, where richer representation information can be extracted since the mask token provides multiple graph maps for the same sample (Yuan et al., 2022).

Fuzzy system learning methods take uncertainty in dynamic environments into consideration to solve transfer learning problems (Shell and Coupland, 2015; Liu et al., 2018). Takagi-Sugeno (TS) and Takagi-Sugeno-Kang (TSK) fuzzy models are two popular rule-based fuzzy systems used in transfer learning (Zuo et al., 2016, 2018a; Deng et al., 2014; Xie et al., 2018). Granular fuzzy regression domain adaptation combines fuzzy system and granular computing, and builds three domain adaptation tasks according to the fuzzy rules and their conclusions in source and target domains (Zuo et al., 2017). Transfer representation learning with TSK fuzzy system extracts representations of original data in fuzzy feature space using fuzzy rules instead of in high-dimensional space using kernel-based nonlinear mapping, and reduces the complexity of data using linear discriminant analysis and principal component analysis, which at the same time can protect the discriminant knowledge and geometric properties of data (Xu et al., 2021a). Transfer learning based on fuzzy residual adopts a residual function to generate target rules from learned source hypothesis, where TSK fuzzy rules are used to describe the marginal distribution of data, which treat target model as a combination of source tasks, making it possible for updating target model in a model-agnostic way (Chen et al., 2022a). Fuzzy multi-output transfer learning considers the shareness and uniqueness of multiple outputs reflected by source fuzzy rules constructed from both output-input dependencies and inter-output correlations, and transfers the learned fuzzy rules to a target domain (Che et al., 2021).

2.7 Summary

Transfer learning is a powerful tool to predict tasks with inadequate information. Existing methods gain much progress in solving different types of transfer learning settings. However, there are still research gaps. For instance, the relationships between multiple source and target domains need further exploration, multiple source domains with different label spaces remain unsolved. Therefore, this thesis aims to fill the mentioned gaps by developing new multi-source transfer learning methods. Motivated by previous studies mentioned in sections 2.1 and 2.2, in chapters 3, 4 and 5, RO 1 and RO 2 are solved by designing new distribution approaches to eliminate the data shift and define the transferable information. The heterogeneous features introduced in section 2.3 are also considered in chapter 4. To extend multi-source transfer learning to more challenging settings introduced in sections 2.4 and 2.5, chapters 6, 7 and 8 solve RO 3 and RO 4 to explore the source-free transfer learning with heterogeneous label spaces. Deep neural networks and fuzzy model introduced in section 2.6 are adopted to construct new models, where RO 5 is achieved. Following chapters introduce our works in details.

Chapter 3

Multi-Source Contribution Learning for Domain Adaptation

3.1 Introduction

As discussed in Chapter 1, many existing transfer learning methods focus on learning one discriminator with single source domain. Sometimes, knowledge from single source domain might not be enough for predicting the target task. For example, in disease diagnosis, hospital A has rich experience and data on children, while hospital B provides data on adults. Learning only from hospital A or B is not enough to obtain a model that can perform well on patients in different ages. Thus, multiple source domains carrying richer transferable information are considered to complete the target task. To fully explore transfer knowledge from multiple source domains, taking advantage of deep learning, convolutional neural networks (CNNs), such as AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016) pre-trained on ImageNet, are widely used to transform source and target data into a latent feature space and extract robust representations (Long et al., 2015; Sun and Saenko, 2016) for visual domain adaptation.

Information from multiple source domains enriches the transfer knowledge compared with that from single source domain. However, multi-source domain adaptation also introduces a major challenge to this research field because of domain shifts, which means we cannot simply combine all source domains as one (Mansour et al., 2009; Redko et al., 2019b). Some methods are developed to tackle multi-source domain adaptation (Li et al., 2018d; Zhao et al., 2018; Zhu et al., 2019a; Liu et al., 2021), while most previous multi-source domain adaptation methods complete a target predictor by averaging all source predictions without due consideration of their different contributions, in other words, how important a source domain is to the target domain. An important source domain indicates it is more similar to target domain and should gain larger combination weight. Different source domains commonly deliver different contributions, which means the combination weights of sources may need to be designed rather than averaging. Although weighted combination is employed (Xu et al., 2018; Peng et al., 2019a; Zhao et al., 2020a), in fact, the weights with minor different quantities might lead to parallel performance as averaged combination, and it might be invalid where the contributions of sources have significant difference, which can result in negative transfer. Negative transfer is a fairly common phenomenon, but identifying when and where it occurs is both difficult and challenging, and there is still no effective way of identifying it.

To measure contributions of multiple sources and reduce the degrading influence of negative transfer which harms the final performance of target predictor, we propose two strategies: a weight learning method with pseudo labels and a fuzzy combination rule for multi-source domain adaptation. The proposed framework adapts all source and target domains simultaneously by minimizing their discrepancies. At the same time, since the target domain might contain diverse characteristics which can be represented by different source domains, the diversities of domains are learned by maximizing their discrepancies. To measure the discrepancy between two domains, both domain-level and class-level discrepancies are considered. Our contributions can be summarized as following:

- Development of a new method to learn weights of source domains using their predicted pseudo labels of target domain. The learned weights are then applied to complete the target predictor, which can take advantage of the best performing source domain. In this way, it will guarantee the target performance if source predictions exhibit significant difference;

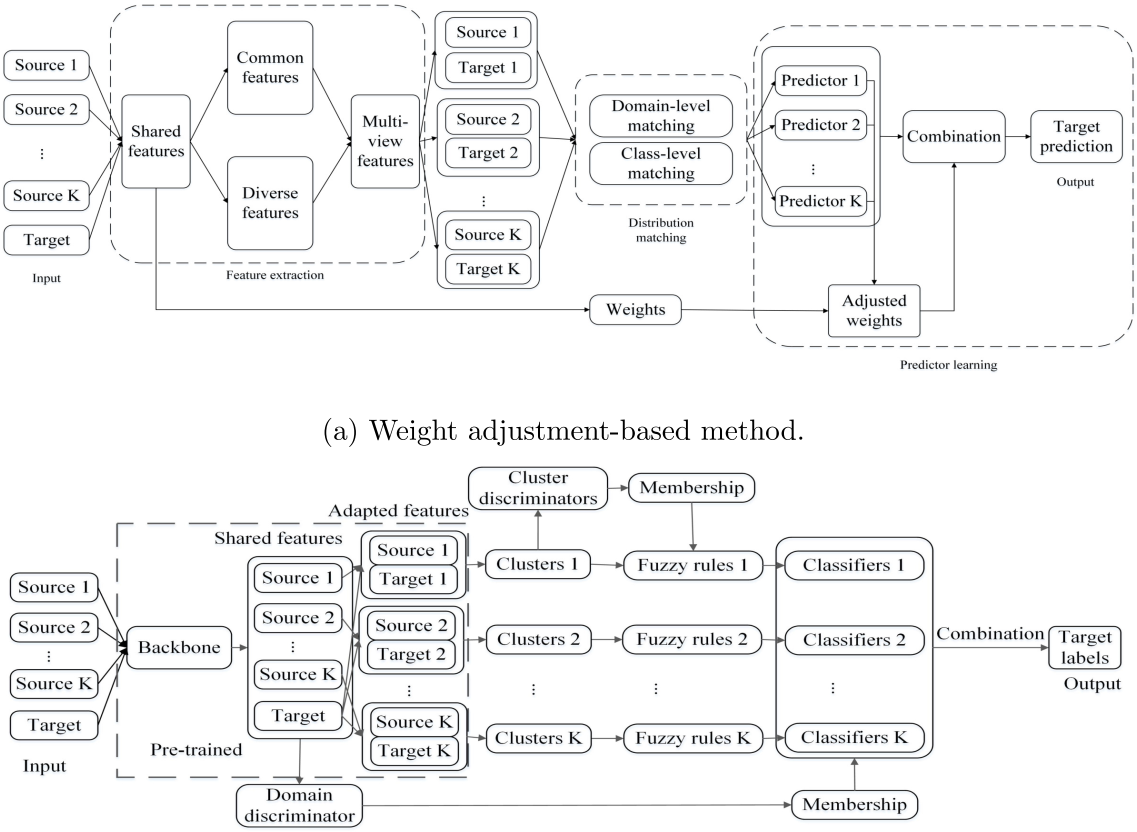
- A representation extraction framework to explore the similarities and the diversities among all source and target domains, which enriches transfer information by providing multiple views of common and specific features. This is valuable when we come to explore target features from multiple aspects and extract comprehensive information, while many existing studies only focus on similarities but ignore diversities;
- An alignment structure to learn the similarities between source and target domains by measuring domain-level and class-level discrepancies simultaneously, which undermines the misalignment of boundary samples. It can enlarge the category distance and reduce the influence of cluster boundaries.
- A fuzzy combination rule for conjoining source classifiers to predict target labels. This is the first study to employ fuzzy membership to define the source contribution to the target task.

3.2 Problem Setting and Notations

We focus on homogeneous unsupervised multi-source domain adaptation, where the feature spaces of labeled source domains and unlabeled target domain have the same dimension. Given K labeled source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$, for each domain $\mathcal{D}_{s_k} = \{(\boldsymbol{X}_{s_k}, Y_{s_k})\} = \{(\boldsymbol{x}_{s_k}^i, \boldsymbol{y}_{s_k}^i)\}_{i=1}^{n_{s_k}}$, where $\boldsymbol{X}_{s_k} \in \mathcal{X}$ represents observed samples which follow distribution $\mathcal{P}_{s_k}$ and $\boldsymbol{Y}_{s_k} \in \mathcal{Y}$ indicates corresponding labels of $\boldsymbol{X}_{s_k}$, $\mathcal{X}, \mathcal{Y}$ indicate original data space and label set, and n_s indicates the number of samples in each source domain. The unlabeled target domain is represented as $\mathcal{D}_t = \{\boldsymbol{X}_t\} = \{\boldsymbol{x}_t^j\}_{j=1}^{n_t}$, where $\boldsymbol{X}_t \in \mathcal{X}$ follows distribution $\mathcal{P}_t$ and n_t is the number of samples. All source and target domains share the same categories, which means the predicted target label $\boldsymbol{Y}_t \in \mathcal{Y}$. We apply the proposed method mainly to image classification.

3.3 The Proposed Multi-Source Contribution Learning Method

The proposed method contains three parts: multi-view feature extraction, multi-level distribution matching and source specific predictors and target predictor learning. The target predictor learning is achieved using two strategies, weight adjustment-based strategy as shown in Fig. 3.1(a), fuzzy rule-based strategy as shown in Fig. 3.1(b). As showed in Fig. 3.1(a), feature extraction entails extracting features that are shared, common and diverse. Shared features are obtained using pre-trained networks which are learned from a very large dataset ImageNet before being divided into common and diverse features (detailed in section 3.3.1). The former represents common knowledge across all domains, the latter specifies knowledge shared by each source and target domains. This approach is expected to express target domain from different perspectives and provide richer information for completing the target task. These extracted multi-view features are then fed into distribution matching, where domain-level matching is employed to adapt source and target features, while class-level matching reduces the misalignment of boundary samples. Source predictors are learned using matched features of source and target domains, while the target predictor is completed by combining source predictors with adjusted weights, which is chosen to reduce negative transfer. In Fig. 3.1(b), target predictor is built by fuzzy rules. Entropy assumption is employed to estimate the similarity of a source sample belonging to a category. By dividing the estimated similarities into different groups, training samples in each source domain are split into multiple clusters, thus fuzzy rules are built to learn new source classifiers. All source classifiers are combined to complete the target task. In order to learn the combination weights, fuzzy membership is estimated using the domain discriminator.



(a) Weight adjustment-based method.

(b) Fuzzy rule-based method.

Figure 3.1 : The whole framework of the proposed method. Shared features are collected using pre-trained networks. Common features represent similarities among all source and target domains, while diverse features represent diversities contained in the target domain which can be expressed by different source domains. Target k means features of target collected using kth source features extraction networks.

3.3.1 Multi-view Feature Extracting

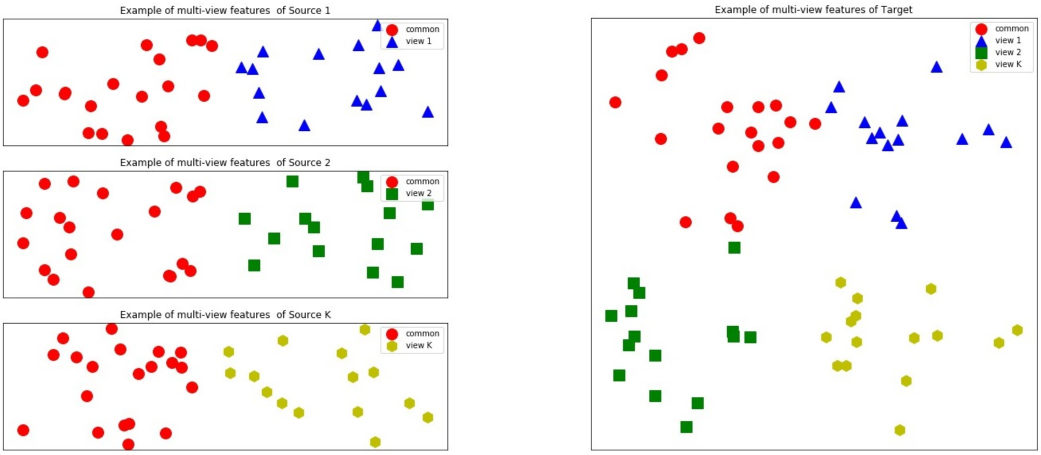
Aiming to extract latent features of source and target domains for adapting, a pre-trained deep neural network ϕ is first used to transform the original data into a shared feature space. The transformation can be expressed as:

$$\begin{gathered}\boldsymbol{f}_{s_k}^i = \phi(\boldsymbol{x}_{s_k}^i, \boldsymbol{\theta}), \\ \boldsymbol{f}_t^j = \phi(\boldsymbol{x}_t^j, \boldsymbol{\theta}), \\ i = 1, 2, \ldots, n_{s_k}, j = 1, 2, \ldots, n_t, k = 1, 2, \ldots, K,\end{gathered} \tag{3.1}$$

where $\boldsymbol{f}_{s_k}, \boldsymbol{f}_t$ represent features in shared feature space, $\boldsymbol{\theta}$ means parameters of deep network ϕ.

Normally a picture shows features from multiple aspects such as context, edge, chrominance, luminance and so on. Based on this fact, the target domain may have multiple aspect characteristics that each view of these characteristics can be reflected by a source domain as being more similar than others. For example, in dataset Office-Home (Venkateswara et al., 2017), domain *Clipart* might resemble domain *Art* more on image text since they all have artistic pictures. At the same time, it might be more similar to domain *Product* on image edge because they are all without background. Fig. 3.2 shows an example of diverse characteristics contained in source and target domains. Assuming each shape in the figure indicates a different characteristic of the target domain, and if this characteristic can be extracted as one kind of feature in latent space, what we expect is to find that characteristic in the source domain which is similar to that from target domain. One source domain might contain partial views, and the union of all source domains could cover more target-like characteristics than any single source domain.

Taking the described factor into consideration, the collected shared features are then split into two parts to represent the target domain more completely. One part carries common transferable information, and the other holds diverse transferable



(a) Source doamin

(b) Target domain

Figure 3.2 : An example of characteristics contained in source and target domains. Common means similar information among all domains, view means the information which is similar between a source and the target but different from other source domains. View k means information of the kth source domain.

information. The common feature extraction can be represented as:

$$\begin{aligned} \boldsymbol{f}^i_{c_{s_k}} &= \phi_{c_k}(\boldsymbol{f}^i_{s_k}, \boldsymbol{\theta}_{c_k}), \\ \boldsymbol{f}^j_{c_{t_k}} &= \phi_{c_k}(\boldsymbol{f}^j_{t}, \boldsymbol{\theta}_{c_k}), \\ i = 1, 2, \ldots, n_{s_k}&, j = 1, 2, \ldots, n_t, k = 1, 2, \ldots, K, \end{aligned} \tag{3.2}$$

while the diverse feature extraction is:

$$\begin{aligned} \boldsymbol{f}^i_{d_{s_k}} &= \phi_{d_k}(\boldsymbol{f}^i_{s_k}, \boldsymbol{\theta}_{d_k}), \\ \boldsymbol{f}^j_{d_{t_k}} &= \phi_{d_k}(\boldsymbol{f}^j_{t}, \boldsymbol{\theta}_{d_k}), \\ i = 1, 2, \ldots, n_{s_k}&, j = 1, 2, \ldots, n_t, k = 1, 2, \ldots, K, \end{aligned} \tag{3.3}$$

where ϕ_{c_k}, ϕ_{d_k} mean feature extractors of kth source domain, and $\boldsymbol{\theta}_{c_k}, \boldsymbol{\theta}_{d_k}$ are corresponding parameters. Each view diverse features can be homogeneous or heterogeneous compared with other views, which means the structures of $\{\phi_{d_k}\}_{k=1}^{K}$ can be different. Besides, ϕ_{c_k} and ϕ_{d_k} can distill redundancy information to some degree by reducing the dimension of shared features $\boldsymbol{f}_{s_k}$ and $\boldsymbol{f}_t$. This dimension reduction is widely used in domain adaptation.

3.3.2 Multi-level Distribution Adapting

The common features extracting processing is controlled by minimizing any discrepancy of common features among all domains, including within source domains and between each source and target domains. Since the target domain has multi-view features, we first adapt source domains from the common view, while the adaptation of sources and target will be done later with the diverse features. Here we choose MMD as the discrepancy measure, measuring the loss function of source common features extraction. It can be written as:

$$\begin{aligned}\mathcal{L}_c =& \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \mathcal{MMD}(\mathcal{D}_{s_{k_1}}, \mathcal{D}_{s_{k_2}}) \\ =& \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\ & \left\| \frac{1}{n_{s_{k_1}}} \sum_{i=1}^{n_{s_{k_1}}} \psi(\boldsymbol{f}^i_{c_{s_{k_1}}}) - \frac{1}{n_{s_{k_2}}} \sum_{j=1}^{n_{s_{k_2}}} \psi(\boldsymbol{f}^j_{c_{s_{k_2}}}) \right\|^2_{\mathcal{H}},\end{aligned} \tag{3.4}$$

where $\|\cdot\|_{\mathcal{H}}$ indicates the reproducing kernel Hillbert space (RKHS) norm, and ψ is kernel-induced feature transformation. During training, the number of samples, $n_{s_{k_1}}$ and $n_{s_{k_2}}$, can be replaced with batch size. This operation is applicable to all MMD calculations in this chapter.

For diverse views, a preferred solution is training a multiple structure networks to extract these features from the target domain directly. However, since the target data is unlabeled, this entirely unsupervised collecting of features for a target task without any assistance rarely meets requirement. Considering this, it can be adapted for extracting diverse features of source domains by maximizing the discrepancy of sources and matching distributions of sources and target simultaneously from diverse views. This can avoid the high correlation between common and diverse features at the same time. The source diverse features extraction loss function is:

$$\begin{aligned}\mathcal{L}_d =&\frac{2}{K(K-1)}\sum_{k_1=1}^{K-1}\sum_{k_2=k_1+1}^{K}\mathcal{MMD}(\mathcal{D}_{s_{k_1}},\mathcal{D}_{s_{k_2}})\\ =&\frac{2}{K(K-1)}\sum_{k_1=1}^{K-1}\sum_{k_2=k_1+1}^{K}\\ &\left\|\frac{1}{n_{s_{k_1}}}\sum_{i=1}^{n_{s_{k_1}}}\psi(\boldsymbol{F}^i_{d_{s_{k_1}}})-\frac{1}{n_{s_{k_2}}}\sum_{j=1}^{n_{s_{k_2}}}\psi(\boldsymbol{F}^j_{d_{s_{k_2}}})\right\|^2_{\mathcal{H}},\end{aligned} \tag{3.5}$$

where $\{\boldsymbol{F}_{d_{s_{k_1}}},\boldsymbol{F}_{d_{s_{k_2}}}\}=\{\boldsymbol{f}_{d_{s_{k_1}}},\boldsymbol{f}_{d_{s_{k_2}}}\}$, if $\boldsymbol{f}_{d_{s_{k_1}}},\boldsymbol{f}_{d_{s_{k_2}}}\in\mathbb{R}^m$, which means homogeneous. $\boldsymbol{F}_{d_{s_{k_1}}}=[\boldsymbol{f}_{d_{s_{k_1}}};\boldsymbol{O}^{m_2}]$ and $\boldsymbol{F}_{d_{s_{k_2}}}=[\boldsymbol{O}^{m_1};\boldsymbol{f}_{d_{s_{k_2}}}]$ if $\boldsymbol{f}_{d_{s_{k_1}}}\in\mathbb{R}^{m_1}$ and $\boldsymbol{f}_{d_{s_{k_2}}}\in\mathbb{R}^{m_2}, m_1\neq m_2$, which means heterogeneous, $\boldsymbol{O}$ is a null matrix. As mentioned before, in this work, we only explore the homogeneous setting but might focus on a heterogeneous setting as future work. Then the total loss of source domains adaptation processing is:

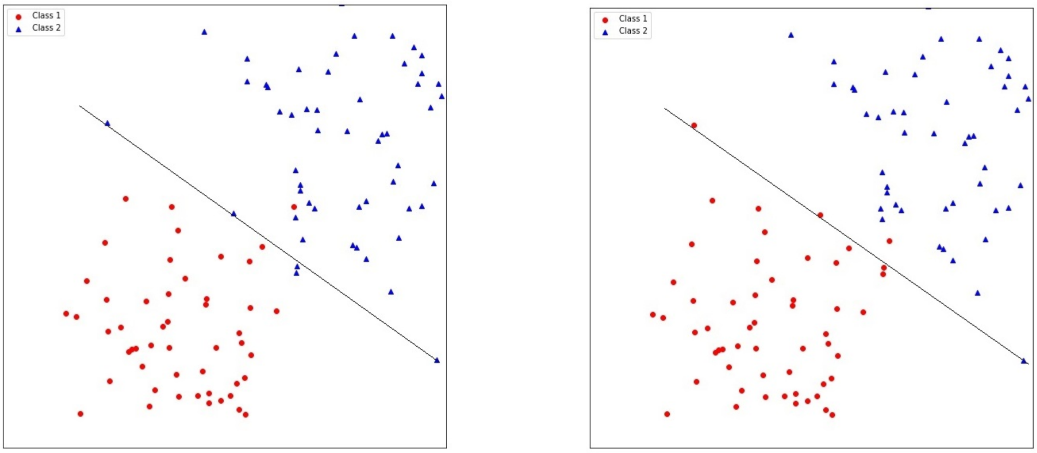
$$\mathcal{L}_s=\mathcal{L}_c-\mathcal{L}_d. \tag{3.6}$$

The extracting of these features for each source domain is controlled by mapping source and target distributions. For each source domain $\mathcal{D}_{s_k}$, the domain-level distribution matching is:

$$\begin{aligned}\mathcal{L}_{domain} =&\mathcal{MMD}(\mathcal{D}_{s_k},\mathcal{D}_t)=\\ &\left\|\frac{1}{n_{s_k}}\sum_{i=1}^{n_{s_k}}\psi(\boldsymbol{F}^i_{cat_{s_k}})-\frac{1}{n_t}\sum_{j=1}^{n_t}\psi(\boldsymbol{F}^j_{cat_{t_k}})\right\|^2_{\mathcal{H}},\end{aligned} \tag{3.7}$$

where $\boldsymbol{F}_{cat_{s_k}}=[\boldsymbol{f}_{c_{s_k}};\boldsymbol{f}_{d_{s_k}}]$, $\boldsymbol{F}_{cat_{d_k}}=[\boldsymbol{f}_{c_{t_k}};\boldsymbol{f}_{d_{t_k}}]$.

Except for adapting each source and target on domain-level, in order to reduce the misalignment of boundary samples, we also consider the class-level distribution matching. A simple synthetic example of boundary samples is given in Fig. 3.3. If the black line is the classifier, samples around it may attract wrong labels. For most complex classification tasks, softmax function is a widely used technology to compute the probabilities of a sample belonging to all classes and to choose the maximal one as its final label. However, samples near class boundaries may get the same probabilities of different classes or a wrong maximal class probability, so we consider maximizing discrepancy among different classes and minimizing the discrepancy within the same classes to solve this problem.



(a) Original classes

(b) Learning classes

Figure 3.3 : An example of boundary samples.

The class-level distribution matching is controlled by:

$$\begin{aligned}\mathcal{L}_{class} =& \frac{1}{C}\sum_{r=1}^{C}\mathcal{MMD}(\mathcal{D}_{s_k}^{r},\mathcal{D}_{t}^{r}) \\ & -(\frac{2\lambda}{3C(C-1)}\sum_{r_1=1}^{C-1}\sum_{r_2=r_1+1}^{C} \\ & (\mathcal{MMD}(\mathcal{D}_{s_k}^{r_1},\mathcal{D}_{s_k}^{r_2})+\mathcal{MMD}(\mathcal{D}_{t}^{r_1},\mathcal{D}_{t}^{r_2}))\big) \\ & -\frac{\lambda}{3C(C-1)}\sum_{r_s=1}^{C}\sum_{r_t\neq r_s}^{C}\mathcal{MMD}(\mathcal{D}_{s_k}^{r_s},\mathcal{D}_{t}^{r_t}).\end{aligned} \tag{3.8}$$

Symbols r, n with superscripts or subscripts in above equations indicate corresponding class index, number of features in each class respectively, C is the total categories of domains, $\lambda \in [0,1]$ is a trade-off constant evaluating the contribution of inter-class discrepancy.

From equation (3.8), if all inter-class discrepancies are calculated, the computation will be high especially when the class number C is large. In practice, the boundary samples are frequently misclassified between two nearest categories. To reduce computation extent, equation (3.8) can be rewritten as below, where we only maximizes the margin of the nearest two classes within each source and target domains:

$$\begin{aligned}\mathcal{L}_{class} =& \frac{1}{C}\sum_{r=1}^{C}\mathcal{MMD}(\mathcal{D}^{r}_{s_k}, \mathcal{D}^{r}_{t})\\ &- \big(\frac{\lambda}{2}(\mathcal{MMD}(\mathcal{D}^{r_{s1}}_{s_k}, \mathcal{D}^{r_{s2}}_{s_k})\\ &+ \mathcal{MMD}(\mathcal{D}^{r_{t1}}_{t}, \mathcal{D}^{r_{t2}}_{t}))\big),\end{aligned} \tag{3.9}$$

where:

$$\begin{aligned}&\mathcal{MMD}(\mathcal{D}^{r}_{s_k}, \mathcal{D}^{r}_{t}) =\\ &\left\| \frac{1}{n^{r}_{s_k}}\sum_{i=1}^{n^{r}_{s_k}}\psi(p^{ir}_{s_k}\cdot \boldsymbol{F}^{ir}_{cat_{s_k}}) - \frac{1}{n^{r}_{t_k}}\sum_{j=1}^{n^{r}_{t_k}}\psi(p^{jr}_{t_k}\cdot \boldsymbol{F}^{jr}_{cat_{t_k}})\right\|^{2}_{\mathcal{H}},\end{aligned} \tag{3.10}$$

$$\begin{aligned}&\mathcal{MMD}(\mathcal{D}^{r_{s1}}_{s_k}, \mathcal{D}^{r_{s2}}_{s_k})) =\\ &\left\| \frac{1}{n^{r_{s1}}_{s_k}}\sum_{i=1}^{n^{r_{s1}}_{s_k}}\psi(p^{ir_{s1}}_{s_k}\cdot \boldsymbol{F}^{ir_{s1}}_{cat_{s_k}}) - \frac{1}{n^{r_{s2}}_{s_k}}\sum_{j=1}^{n^{r_{s2}}_{s_k}}\psi(p^{jr_{s2}}_{s_k}\cdot \boldsymbol{F}^{jr_{s2}}_{cat_{s_k}})\right\|^{2}_{\mathcal{H}},\end{aligned} \tag{3.11}$$

$$\begin{aligned}&\mathcal{MMD}(\mathcal{D}^{r_{t1}}_{t}, \mathcal{D}^{r_{t2}}_{t}) =\\ &\left\| \frac{1}{n^{r_{t1}}_{t_k}}\sum_{i=1}^{n^{r_{t1}}_{t_k}}\psi(p^{ir_{t1}}_{t_k}\cdot \boldsymbol{F}^{ir_{t1}}_{cat_{t_k}}) - \frac{1}{n^{r_{t2}}_{t_k}}\sum_{j=1}^{n^{r_{t2}}_{t_k}}\psi(p^{jr_{t2}}_{t_k}\cdot \boldsymbol{F}^{jr_{t2}}_{cat_{t_k}})\right\|^{2}_{\mathcal{H}}.\end{aligned} \tag{3.12}$$

p is the probability of a sample belonging to class r, subscript $s1, t1$ indicate the maximal class probabilities, $s2, t2$ represent the second maximal class probabilities. The total loss of target domain adaptation is :

$$\mathcal{L}_t = \mathcal{L}_{domain} + \mathcal{L}_{class}. \tag{3.13}$$

3.3.3 Predictions Learning Based on Weight Adjustment

After adapting all source and target domains, the predictors of source tasks can be learned and applied to the target task. Cross entropy is employed to optimize the predictors, for each source domain $\mathcal{D}_k$, it can be represented as:

$$\mathcal{L}_p = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} y_{s_k}^i \log\big(P_{s_k}(\boldsymbol{F}_{cat_{s_k}}^i)\big), \tag{3.14}$$

P_{s_k} is the predictor of kth source domain. When applying the learned source predictors to the target task, it is desired that all source predictors could return the same results as the same target samples. So the cross-domain constraint is added to minimize errors of different predictions on the same target samples:

$$\begin{aligned}\mathcal{L}_{cro} = \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\ \big(\frac{1}{n_t} \sum_{j=1}^{n_t} \big|P_{s_{k_1}}\big(\boldsymbol{F}_{catt_{k_1}}^j\big) - P_{s_{k_2}}\big(\boldsymbol{F}_{catt_{k_2}}^j\big)\big|\big).\end{aligned} \tag{3.15}$$

The total loss of predictor learning of each source is:

$$\mathcal{L} = \mathcal{L}_p + \alpha\mathcal{L}_s + \beta\mathcal{L}_t + \gamma\mathcal{L}_{cro}, \tag{3.16}$$

α, β, γ are trade-off parameters.

To complete target task with multiple source predictions, a weights learning method is developed to evaluate the contributions of sources. Many previous studies weigh source predictions using average mean method or by normalizing similarities based on distribution distance. A simple and usual similarity learning is:

$$\begin{aligned} \omega_{sim}^{k} &= \frac{1}{Dis(\mathcal{D}_{s_k}, \mathcal{D}_t)}, \\ \omega_{s_k} &= \frac{\omega_{sim}^{k}}{\sum_{k'=1}^{K} \omega_{sim}^{k'}}, \end{aligned} \tag{3.17}$$

$Dis(\mathcal{D}_{s_k}, \mathcal{D}_t)$ means the distribution distance of kth source and target domains in shared feature space. In our work, it is:

$$\begin{aligned} &Dis(\mathcal{D}_{s_k}, \mathcal{D}_t) = \mathcal{MMD}(\mathcal{D}_{s_k}, \mathcal{D}_t) = \\ &\qquad \left\| \frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \psi(\boldsymbol{f}_{s_k}^{i}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \psi(\boldsymbol{f}_t^{j}) \right\|_{\mathcal{H}}^{2}. \end{aligned} \tag{3.18}$$

The corresponding target prediction can be expressed as:

$$\boldsymbol{y}_t = \sum_{k=1}^{K} \omega_{s_k} \cdot P_{s_k}(\boldsymbol{F}_{cat_{t_k}}). \tag{3.19}$$

This method indeed yields larger weights of the more similar source predictions. However, if the source performances have obvious differences, the minor disparities of weight values may fail to return preferable results on the target. To increase the disparities between source weights, we add an adjusting constant controlled by prediction labels to adjust the weight values. By doing this, the closest source domain is expected to dominate the prediction of target samples. As mentioned in equation (3.15), cross-domain constraint is used to ensure that the multiple source predictors could return the same label of the same target sample. Hence, weight adjustment mainly affects the target samples that are predicted differently by the source classifiers. For those samples, strengthening the importance of the target labels returned by the closest source domain and weakening those returned by the furthest source domains could guarantee that we get the correct labels with high probability.

It is assumed that the same predicted pseudo target labels returned by source predictors are “correct labels”. These “correct labels” are used to decide when the original source weights ω_{s_k} should be adjusted. Since the prediction learning is processed based on batches and not the whole dataset, the threshold value of "correct labels" is set as a, while the number of target samples in every iteration is b. The threshold means the source classifiers perform quite stably on the target domain, indicating that there is no need to adjust weights over each sample, which might be time-consuming.

Pseudo labels returned by each source predictor is:

$$\boldsymbol{y}_{t_k} = P_{s_k}(\boldsymbol{F}_{cat_{t_k}}). \tag{3.20}$$

Then the "correct labels" can be expressed as:

$$\begin{aligned} \boldsymbol{y}_{tc} &= \mathcal{Z}(\{\boldsymbol{y}_{t_k}\}_{k=1}^{K}), \\ n_{tc} &= \mathcal{C}(\boldsymbol{y}_{tc}), \end{aligned} \tag{3.21}$$

where $\mathcal{Z}$ is the operation to get the same predicted labels, $\mathcal{C}$ means function to count the number n_{tc} of "correct labels". When $n_{tc} >= a$, the source weights in equation (3.17) can be rewritten as:

$$\begin{aligned} \omega_{sk} &= R(G(\omega_{s_k} + (1 - a/b))), \\ \omega_{s_k} &= \omega_{sk} + \frac{a}{K \cdot b} + \frac{(K-2) \cdot a}{K(K-1) \cdot b}, \\ if.\omega_{s_k} &= \max[\omega_{s_1}, \omega_{s_2}, \cdots, \omega_{s_K}], \\ \omega_{sk} &= R(G(\omega_{s_k} - (1 - a/b))), \\ \omega_{s_k} &= \omega_{sk} - \frac{a}{K \cdot b}, \\ if.\omega_{s_k} &= \min[\omega_{s_1}, \omega_{s_2}, \cdots, \omega_{s_K}], \\ \omega_{sk} &= R(G(\omega_{s_k})), \\ \omega_{s_k} &= \omega_{sk} - \frac{a}{K(K-1) \cdot b}, \end{aligned} \tag{3.22}$$

where G is sigmoid function, R is normalized function $R = \frac{\omega_{sk}}{\sum_{sk'=1}^{K} \omega_{sk'}}$, ω_{s_k} satisfy $\sum_{k=1}^{K} \omega_{s_k} = 1$. Apply the above new weights to equation (3.19) when n_{tc} is larger than threshold, target labels can be predicted. The whole processing is described in Algorithm 1.

3.3.4 Predictions Learning Based on Fuzzy Rules

The Takagi–Sugeno fuzzy model is a popular fuzzy architecture. For data pair $(\boldsymbol{x}, \boldsymbol{y})$, the rule is:

$$\text{if } \boldsymbol{x} \ is \ A_m, \text{then } \boldsymbol{y} \ is \ P_m(\boldsymbol{x}), m = 1, 2, \cdots, M. \tag{3.23}$$

A_m is the fuzzy set of the mth rule, P_m is the corresponding output function. The output of the fuzzy system is expressed as:

$$\boldsymbol{y} = \sum_{m=1}^{M} p_m \cdot P_m(\boldsymbol{x}) \tag{3.24}$$

p_m is the membership of data belonging to a set.

In the classification task, the classifier can identify an item in different views, for example, front view, partial view, rotate view and so on. It cannot distinguish the different views of the item but only "remembers" its features during learning. The information level of the same item in different views is actually different, and samples with the same level information are more similar to each other compared with those with different level information. Hence, according to the information level, to construct a fuzzy model for classification, we divide the samples into multiple groups to learn the multiple classifiers of each source domain, which is expected to benefit the classification.

Using the estimated similarity to represent the information level contained in a sample, the similarity of each sample belonging to the class in kth source domain can be estimated by the classifier:

$$p_{s_k} = \max(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k})))) = \max(P_{s_k}(\boldsymbol{F}_{cat_{s_k}})), p_{s_k} \in [0, 1] \tag{3.25}$$

where $\phi_k = (\phi_{c_k}, \phi_{d_k})$.

Algorithm 1 Weight adjustment-based multi-source domain adaptation

```
**Input:** Source domains $\{\mathcal{D}_{s_k}\}_{k=1}^K$, target domain $\mathcal{D}_t$, training iteration $\mathcal{I}$, pre-trained model $\phi(\cdot, \boldsymbol{\theta})$;
**Initialization:** Feature extraction networks $\{\phi_{c_k}(\cdot, \boldsymbol{\theta}_{c_k})\}_{k=1}^K$, $\{\phi_{d_k}(\cdot, \boldsymbol{\theta}_{d_k})\}_{k=1}^K$, and source predictors $\{P_{s_k}\}_{k=1}^K$;
**for** $\epsilon = 1$, $\epsilon < \mathcal{I}$, $\epsilon + +$, **do**
    $\{(\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k})\}_{k=1}^K \leftarrow$ collect $m$ batch pairs from corresponding $\mathcal{D}_{s_k}$ randomly;
    $\{\boldsymbol{x}_t\} \leftarrow$ collect $m$ batch pairs from $\mathcal{D}_t$ randomly;
    $\{\boldsymbol{f}_{s_k}, \boldsymbol{f}_t\}_{k=1}^K \leftarrow \{\phi(\{\boldsymbol{x}_{s_k}, \boldsymbol{x}_t\}, \boldsymbol{\theta})\}_{k=1}^K$, collect shared features according to (3.1);
    $\{\boldsymbol{f}_{c_{s_k}}, \boldsymbol{f}_{c_{t_k}}\}_{k=1}^K \leftarrow \{\phi_{c_k}(\{\boldsymbol{f}_{s_k}, \boldsymbol{f}_t\}, \boldsymbol{\theta}_{c_k})\}_{k=1}^K$, collect common features according to (3.2);
    $\{\boldsymbol{f}_{d_{s_k}}, \boldsymbol{f}_{d_{t_k}}\}_{k=1}^K \leftarrow \{\phi_{d_k}(\{\boldsymbol{f}_{s_k}, \boldsymbol{f}_t\}, \boldsymbol{\theta}_{d_k})\}_{k=1}^K$, collect diverse features according to (3.3);
    $\mathcal{L}_s \leftarrow \mathcal{L}_c - \mathcal{L}_d$, compute loss within source domains according to (3.4), (3.5) and (3.6);
    $\mathcal{L}_t \leftarrow \mathcal{L}_{domain} + \mathcal{L}_{class}$, compute loss between source and target domains according to (3.7), (3.9) and (3.13);
    Compute prediction loss $\mathcal{L}_p$ according to (3.14);
    Compute cross-domain constrain loss $\mathcal{L}_{cro}$ according to (3.15);
    Compute total loss $\mathcal{L}$ according to (3.16);
    Compute $\omega_{s_k}$ according to (3.17);
    $\{\boldsymbol{y}_{t_k}\}_{k=1}^K \leftarrow \{P_{s_k}(\boldsymbol{F}_{cat_{t_k}})\}_{k=1}^K$, collect pseudo labels according to (3.20);
    $\boldsymbol{y}_{tc} \leftarrow \mathcal{Z}(\{\boldsymbol{y}_{t_k}\}_{k=1}^K)$, $n_{tc} \leftarrow \mathcal{C}(\boldsymbol{y}_{tc})$, collect the same labels according to (3.21);
    **if** $n_{tc} >= a$ **then**
        Adjust $\omega_{s_k}$ according to (3.22);
    **end if**
    $\boldsymbol{y}_t \leftarrow \sum_{k=1}^K \omega_{s_k} \cdot P_{s_k}(\boldsymbol{F}_{cat_{t_k}})$, return target labels according to (3.19)
    Update $\phi(\cdot, \boldsymbol{\theta})$, $\{\phi_{c_k}(\cdot, \boldsymbol{\theta}_{c_k})\}_{k=1}^K$, $\{\phi_{d_k}(\cdot, \boldsymbol{\theta}_{d_k})\}_{k=1}^K$, and source predictors $\{P_{s_k}\}_{k=1}^K$;
**end for**
**Output:** Predicted target label $\boldsymbol{y}_t$.
```

Divide the closed interval $[0,1]$ into M sub-intervals, $[0,a_1)$, $\cdots$, $[a_{k-1},a_k)$, $\cdots$, $[a_{M-1},1]$, the source samples are split into different clusters according to the value of the estimated similarity. For the mth cluster, a classifier $P_{s_{km}}$is trained by minimizing the cross-entropy loss:

$$\mathcal{L}_m = -\frac{1}{n_{s_{km}}}\sum_{i=1}^{n_{s_{km}}} \boldsymbol{y}_{s_k}^i \log(P_{s_{km}}(\boldsymbol{F}_{cat_{s_k}}^i)). \tag{3.26}$$

$n_{s_{km}}$ is the number of cluster samples.

A cluster discriminator is trained using samples from each cluster to estimate the membership of new inputs. The cluster discriminator of the kth source domain P_{c_k} is parameterized by:

$$\mathcal{L}_{Pc} = -\frac{1}{n_{s_k}}\sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{c_k}^i \log(P_{c_k}(\boldsymbol{F}_{cat_{s_k}}^i)). \tag{3.27}$$

$\boldsymbol{y}_{c_k}$ is cluster label. The membership vector is:

$$\boldsymbol{p}_{c_k} = P_{c_k}(\boldsymbol{F}_{cat_{s_k}}). \tag{3.28}$$

The fuzzy model for each source domain in equations (3.23)-(3.24) can be re-written as:

$$\text{if } \boldsymbol{x}_{s_k} \ is\ A_m, \text{then } \boldsymbol{y}_{s_k} \ is\ P_{s_{km}}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), m = 1,2,\cdots,M. \tag{3.29}$$

The prediction of kth source domain is expressed as:

$$\boldsymbol{y}_{s_k} = \boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))) = \boldsymbol{p}_{c_k}^T \cdot \begin{bmatrix} P_{s_{k1}}(\boldsymbol{F}_{cat_{s_k}}) \\ \cdots \\ P_{s_{kM}}(\boldsymbol{F}_{cat_{s_k}}) \end{bmatrix}, \tag{3.30}$$

Cross-entropy loss of $\boldsymbol{P}_{s_k}$ is:

$$\mathcal{L}_{Pf} = -\frac{1}{n_{s_k}}\sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\boldsymbol{F}_{cat_{s_k}}^i)). \tag{3.31}$$

As in equation (3.15), the cross-domain constraint for learning source classifiers in the proposed fuzzy model is re-written as:

$$\mathcal{L}_{crof} = \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \Big(\frac{1}{n_t} \sum_{j=1}^{n_t} \big| \boldsymbol{p}_{c_{k_1}}^T \cdot \boldsymbol{P}_{s_{k_1}}(\boldsymbol{F}_{cat_{t_{k_1}}}^j) - \boldsymbol{p}_{c_{k_2}}^T \cdot \boldsymbol{P}_{s_{k_2}}(\boldsymbol{F}_{cat_{t_{k_2}}}^j) \big| \Big). \tag{3.32}$$

The loss of learning the fuzzy rule-based source classifier is:

$$\boldsymbol{P}_{s_k} = \mathop{\arg\min}_{\substack{\boldsymbol{P}_{s_k} \in \mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \sim \mathcal{D}_{s_k}}} \sum_{m=1}^{M} \mathcal{L}_m + \lambda_1 \mathcal{L}_{Pf} + \lambda_2 \mathcal{L}_{crof}. \tag{3.33}$$

To complete the target task, all source classifiers are combined to predict the target labels, which can be expressed as a fuzzy model:

$$\text{if } \boldsymbol{x}_t \textit{ is } \mathcal{D}_{s_k}, \text{then } \boldsymbol{y}_t \textit{ is } \boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\phi_k(\phi(\boldsymbol{x}_t))), k = 1, 2, \cdots, K. \tag{3.34}$$

The final prediction of the target data is:

$$\boldsymbol{y}_t = \boldsymbol{p}_d^T \cdot \begin{bmatrix} \boldsymbol{p}_{c_1}^T \cdot \boldsymbol{P}_{s_1}(\phi_k(\phi(\boldsymbol{x}_t))) \\ \cdots \\ \boldsymbol{p}_{c_K}^T \cdot \boldsymbol{P}_{s_K}(\phi_k(\phi(\boldsymbol{x}_t))) \end{bmatrix}, \tag{3.35}$$

$\boldsymbol{p}_d$ is the membership vector, indicating the probability of the target samples belonging to a source domain.

To define the membership, pseudo label-based and feature-based strategies are used to determine the combination rule. First, source classifiers directly pseudo label the target data, noting the number of target samples which obtain the same results from multiple source classifiers in each batch as n_c, batch size as n_b, the frequency of $n_c = n_b$ is a_c, and a threshold a is defined to identify if there is a significant difference among the predictions. If $a_c > a$, it means multiple source domains contribute similarly to the target domain, the averaged combination is then used, the element value of $\boldsymbol{p}_d$ is $\frac{1}{K}$, if $a_c \leq a$, a domain discriminator is used to estimate the element values.

We collected the shared features $\{\boldsymbol{f}_{s_k}\}_{k=1}^{K}$ and $\boldsymbol{f}_t$, the domain discriminator P_d is controlled by:

$$\mathcal{L}_{Pd} = -\frac{1}{n_s}\sum_{i=1}^{n_s} \boldsymbol{y}_d^i \log(P_d(\boldsymbol{f}_s^i)). \tag{3.36}$$

$\boldsymbol{y}_d$ is domain label, $\boldsymbol{f}_s = \bigcup_{k=1}^{K}\{\boldsymbol{f}_{s_k}\}$, $n_s = \sum_{k=1}^{K} n_{s_k}$. The membership vector is:

$$\boldsymbol{p}_d = P_d(\boldsymbol{f}_t), \boldsymbol{p}_d = [p_{d_1}, \cdots, p_{d_K}]^T. \tag{3.37}$$

The combination rule of target classifier can be formulated as:

$$\boldsymbol{y}_t = \begin{cases} \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\phi_k(\phi(\boldsymbol{x}_t))), \text{if } a_c > a, \\ \sum_{k=1}^{K} p_{d_k} \cdot (\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\phi_k(\phi(\boldsymbol{x}_t))), \text{if } a_c \leq a, \end{cases} \tag{3.38}$$

$$k = 1, 2, \cdots, K.$$

The whole processing is described in Algorithm 2.

Algorithm 2 Fuzzy rule-based multi-source domain adaptation

```
**Input:** Source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$, target domain $\mathcal{D}_t$, training iteration $\mathcal{I}$, pre-trained source classifier $P_{s_k}$, feature extractors $\phi_k, \phi$;
**Initialization:** Cluster discriminator $P_{c_k}$, domain discriminator$P_d$, fuzzy rules $\boldsymbol{P}_{s_k}$;
Build fuzzy sets according to (3.25)
Train cluster discriminator according to (3.27);
**for** $\epsilon = 1$, $\epsilon < \mathcal{I}$, $\epsilon + +$, **do**
    Build fuzzy rules as in (3.28), (3.29) and (3.29);
    Compute loss of fuzzy rule-based classifier according to (3.33);
    Update fuzzy rules $\boldsymbol{P}_{s_k}$;
**end for**
Train domain discriminator according to (3.36);
Compute target membership according to (3.37);
**Output:** Predicted target label $\boldsymbol{y}_t$.
```

3.4 Experiments

In this section, we apply the proposed method to some popular real-world visual datasets for multiple sources domain adaptation classification tasks. Results, comparison, and analysis will be provided.

3.4.1 Datasets and Baselines

Experimental datasets include Office-31, ImageCLEF-DA, Office-Home and Office-Caltech10.

Office-31 is an unbalanced dataset comprising 4110 images from datasets Amazon (A), Webcam (W) and DSLR (D) which share 31 categories, and each dataset is regarded as a domain. Amazon contains 2817 images, Webcam has 795 images and DSLR holds 498 images. The number of images in each category is different. Proposed method is tested via building three tasks: $A, W \to D$; $A, D \to W$; $D, W \to A$.

ImageCLEF-DA is a balanced dataset containing 1800 images from datasets Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P) which share 12 categories, each domain corresponding to a dataset. Every category contains 50 images and there are 600 images in each domain. Proposed method is tested via building three tasks: $I, C \to P$; $I, P \to C$; $C, P \to I$.

Office-Home is a new and large unbalanced dataset consisting of 15588 images from datasets Art (A), Clipart (C), Product (P) and Real World (R) which share 65 categories. Art has 2427 images, Clipart contains 4365 images, Product comprises 4439 images, and Real World holds 4357 images. Treating each dataset as a domain, the proposed method is tested via building four tasks: $A, C, P \to R$; $A, C, R \to P$; $A, P, R \to C$, $C, P, R \to A$.

Office-Caltech10 is an unbalanced dataset extended by Office-31 and Caltech, which consists 2533 images sharing 10 categories. Caltech (C) contains 1123 images, Amazon (A) contains 958 images, Webcam (W) holds 295 images, and DSLR (D) has 157 images. Treating each dataset as a domain, proposed method is tested via building four tasks: $A, D, W \to C$; $C, D, W \to A$; $A, C, D \to W$, $A, C, W \to D$.

There are three standards: "Single best", "Source Combine" and "Multi-Source". "Single best" means the best performance of single source domain using the single source domain adaptation method, "Source Combine" is performance returned by a single source domain adaptation method with multiple sources, which unites all source domains as one, "Multi-Source" is domain adaptation with multiple sources, all methods complete target task using different combination rules. Comparable state-of-the-art domain adaptation methods are as follows. The single source domain adaptation methods include:

- DAN: Deep adaptation network (Long et al., 2015);
- RevGrad: Reverse gradient (Ganin and Lempitsky, 2015);
- D-CORAL: Correlation alignment for domain adaptation (Sun and Saenko, 2016);
- MRAN: Multi-representation adaptation network (Zhu et al., 2019b);
- MDDA: Manifold dynamic distribution adaptation (Wang et al., 2020b);
- DDAN: Dynamic distribution adaptation network (Wang et al., 2020b).
- MADA: Multi-adversarial domain adaptation (Pei et al., 2018);
- DAAN: Dynamic adversarial adaptation network (Yu et al., 2019a);
- ADDA: Adversarial discriminative domain adaptation (Tzeng et al., 2017);

- CyCADA: Cycle-consistent adversarial domain adaptation (Hoffman et al., 2018);

The multi-source domain adaptation methods include:

- DCTN: Deep cocktail network (Xu et al., 2018);
- M3SDA: Moment matching for multi-source domain adaptation (Peng et al., 2019a);
- MFSAN: Multiple feature spaces adaptation network (Zhu et al., 2019a);
- DFRE: Distribution fusion and relationship extraction network (Li et al., 2020b).

All results for comparison are collected from previous studies based on $ResNet$, except for MFSAN on datasets Office-Home and Office-Caltech10, and we ran them ourselves using code released by authors [*].

3.4.2 Parameter Setting and Effect of Different Similarity Metrics

Our experiments were performed using Pytorch based on $ResNet50$ (shared network). Followed by feature extraction networks $\phi_{c_k}(\cdot, \boldsymbol{\theta}_{c_k})$ and $\phi_{d_k}(\cdot, \boldsymbol{\theta}_{d_k})$ have 3 convolution layers, the source predictors P_{s_k} contains 1 fully connected layer. We fine-tune all convolutional layers using back-propagation with Stochastic Gradient Descent (SGD), the momentum is 0.9, the learning rate η follows the same strategy in (Ganin and Lempitsky, 2015), that is $\eta = \frac{\eta_0}{(1+10p)^{0.75}}$, where $\eta_0 = 0.01$, p is the training progress changing linearly from 0 to 1. Learning rate of shared network is one tenth of other layers. Batchsize $b = 32$, trade-off parameters $\lambda = 0.01$, α, β, γ

[*] https://github.com/easezyc/deep-transfer-learning

follow the existing work (Zhu et al., 2019a), that is $\alpha = \beta = \gamma = \frac{2}{1+exp(-10p')} - 1$, where p' changes from 0 to 1 linearly.

Threshold value a is defined by target task and varies accordingly. This also determines when the adjustment of weights should be started. We will explain how to choose an appropriate threshold value. Fig. 3.4 shows the source training losses of one experiment using the proposed method, and Fig. 3.5 displays the test accuracy of experiments with and without using threshold.

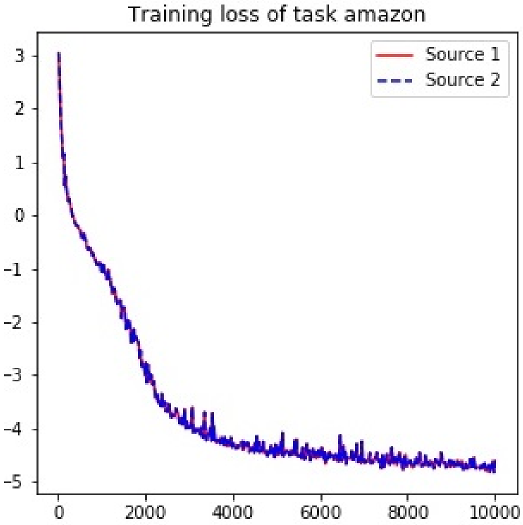
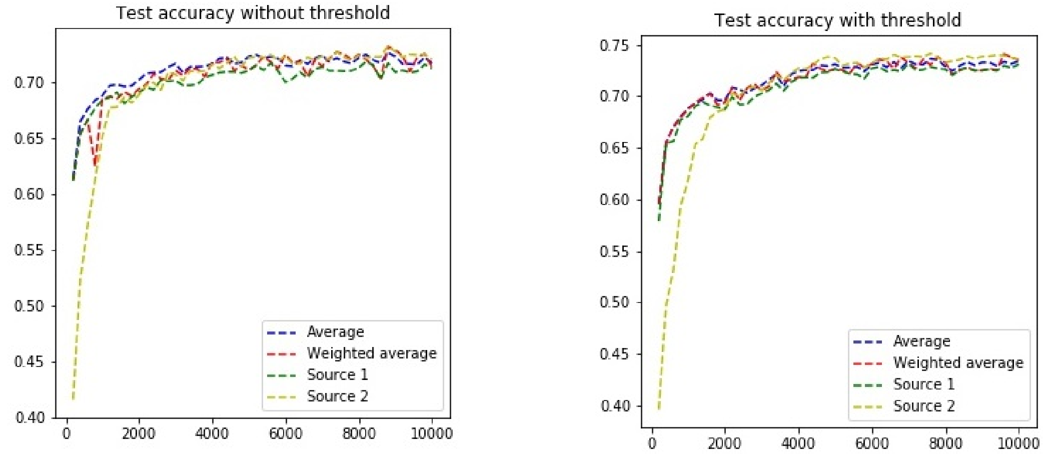


Figure 3.4 : Training loss of the proposed method. Taking task Amazon in Office-31 as an example.

It can be seen that early period of the training process (below 2000 times), the training loss of each source reduces sharply. Combining with Fig. 3.5, the test accuracy on task increases markedly. If we adjust all weights without the control of threshold, the accuracy of multi-source domain adaptation is near to single source performance from the very beginning of training. But at that time, all predictions of single source domain is not yet convergent, which means they cannot perform well on the target domain, thus giving a very large weight of a source which can harm the performance. It is appropriate to adjust weights when the training losses of



(a) Without threshold

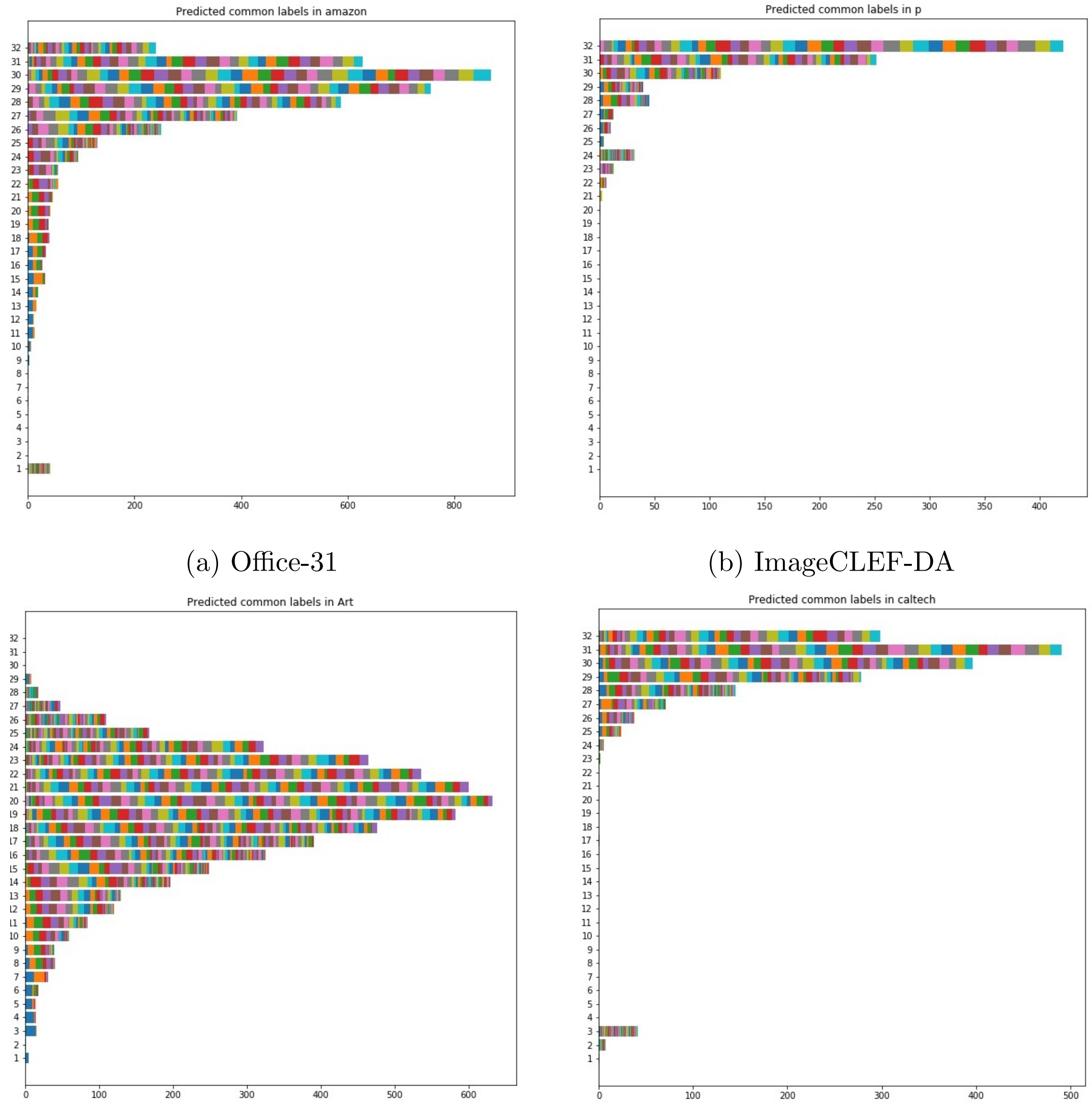
(b) With threshold

Figure 3.5 : Test accuracy of the proposed method without and with using threshold. Taking task Amazon in Office-31 as an example.

sources start to reduce slowly. Changing weights by observing loss reduction during experiments might be inconvenient to operate, so we turn to observe the same target labels returned by source domains.

Fig. 3.6 shows the same target labels returned by source predictors in every batch. As trainning progresses, the number of the same target labels n_{tc} increases. For small datasets, this number falls largely in the interval between 29 to 32, for large dataset, the value fluctuates mainly between 18 to 22, for datasets with a medium number of samples, the value falls between 27 to 30. The most frequent number means the performance of single source starts to become stable. In other words, adjustment of weights should be started before the occurrence of these values. A small threshold value means that the adjustment of weights starts early, while a large one means that in most cases, there is no need to adjust the weights. Normally, the larger the n_{tc} is, the less will be the differences among single source predictions.

Fig. 3.7 shows test accuracy with different threshold values. The performance of



(a) Office-31

(b) ImageCLEF-DA

(c) Office-Home

(d) Office-Caltech10

Figure 3.6 : Number of same labels returned by source predictors. Taking target domain Amazon as example for Office-31, Pascal VOC 2012 for ImageCLEF-DA, Art for Office-Home and Caltech for Office-Caltech10.

final target predictor changes with different thresholds. If the threshold is too large compared with the most frequently occurring number of the same labels (rarely adjust weights), the accuracy is reduced. When dealing with a large quantity of experiments if we find the threshold of each different task, for convenience, instead of learning specific a for each target domain in every dataset, we set $a = 30$ for small datasets ImageCLEF-DA and Office-Caltech10, $a = 24$ for dataset Office-31, $a = 20$ for large dataset Office-Home.

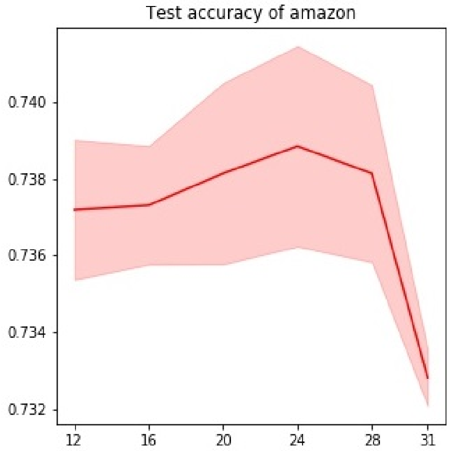


Figure 3.7 : Test accuracy of the proposed method with different threshold values. Taking task Amazon in Office-31 as an example, the accuracy is average result of three times experiments. The red line represents accuracy while the light red area signifies standard deviation.

In this work, we choose MMD as a similarity metric to measure the distance between two distributions. To evaluate the effectiveness of MMD, experiments based on another popular discrepancy measurement named Wasserstein distance (WD) are taken as a comparison on dataset Office-31. The source order is the same as described, for example, $S1$ is domain A while $S2$ is domain W in task $A, W \to D$. "S" means single source and "M" means multi-source. Experiments are repeated for three times. Table. 3.1 indicates that the model trained with MMD outperforms the model trained with WD, it achieves higher accuracy of both single source and multi-source domain adaptation.

Table 3.1 : Accuracy (%) with different similarity metrics.

Standards		A, W→D	A, D→W	W, D→A	Avg
WD	S1	96.1	98.0	71.6	89.4
	S2	99.7	98.6	72.4	
	M	99.7	98.6	72.2	90.2
MMD	S1	96.8	98.1	73.3	90.1
	S2	99.7	98.8	**74.0**	
	M	**99.8**	**98.9**	73.9	**90.9**

3.4.3 Results and Analysis Based on Weight Adjustment

For each dataset, we run the proposed method five times with random initialized parameters and return the average performance. Tables 3.2, 3.3, 3.4 and 3.5 show results of the proposed and compared methods on Office-31, ImageCLEF-DA, Office-Home and Office-Caltech10, respectively. It can be seen that the proposed method outperforms other state-of-the-art domain adaptation methods, and obtains the highest accuracy on most target tasks.

In general, domain adaptation with multiple source domains shows superior results compared with single best results. That means multiple sources with richer transferable information have positive influence on target task. At the same time, multi-source domain adaptation with combination rules performs better than simply combining all source domains as one. Simply combining them fails to consider the specific knowledge contained in each source domain, and transforms features of all domains into a common latent feature space. However, sometimes, a feature space that can adapt all domain distributions may not exist, which means the predictions learned based on these features in that same space may not work as well on both source and target domains as desired. Multi-source domain adaptation with combination rules, on the contrary, explores common features as well as specific features, and learn specific predictors of source domains, by which the distributions of source and target domains can be better matched.

Table 3.2 : Comparison of classification accuracy (%) on dataset Office-31

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Single best	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGard	99.1	96.9	68.2	88.1
	MADA	99.6	97.4	70.3	89.1
	MRAN	**99.8**	96.9	70.9	89.2
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGard	99.7	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	MSCLDA	**99.8**	**98.8**	**73.7**	**90.8**

Tables 3.6, 3.7, 3.8 and 3.9 present classification accuracy on Office-31, ImageCLEF-DA, Office-Home and Office-Caltech10 with different combination rules and different distribution matching strategies. Since the results of averaged combination and of

Table 3.3 : Comparison of classification accuracy (%) on dataset ImageCLEF-DA

Standards	Method	I, C→P	I, P→C	P, C→I	Avg
Single best	ResNet	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGard	75.0	**96.2**	87.0	86.1
	DAAN	78.5	94.3	91.3	88.0
	MADA	75.2	96.0	88.8	86.7
	MRAN	78.8	95.0	93.5	89.1
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGard	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN	79.1	95.4	93.6	89.4
	MSCLDA	**79.5**	95.9	**94.3**	**89.9**

Table 3.4 : Comparison of classification accuracy (%) on dataset Office-Home

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Single best	ResNet	75.4	79.7	49.6	65.3	67.5
	DAN	75.9	80.3	56.5	68.2	70.2
	D-CORAL	76.3	80.3	53.6	67.0	69.3
	RevGard	75.8	80.4	55.9	67.9	70.0
	DAAN	74.0	78.8	54.0	66.3	68.3
	MRAN	77.5	**82.2**	60.0	70.4	72.5
Source Combine	DAN	82.5	79.0	59.4	68.5	72.4
	D-CORAL	**82.7**	79.5	58.6	68.1	72.2
	RevGard	**82.7**	79.5	59.1	68.4	72.4
Multi-Source	MFSAN	80.8	79.0	60.7	70.0	72.6
	MSCLDA	80.6	79.9	**61.4**	**71.6**	**73.4**

Table 3.5 : Comparison of classification accuracy (%) on dataset Office-Caltech10

Standards	Method	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
Single best	ResNet	82.5	91.2	98.9	99.2	93.0
	ADDA	88.8	94.5	99.1	98.0	95.1
	CyCADA	89.7	96.2	98.9	97.3	95.5
Source Combine	DAN	89.7	94.8	99.3	98.2	95.5
	ADDA	90.2	95.0	99.4	98.2	95.7
	CyCADA	91.0	95.9	99.0	97.8	95.9
Multi-Source	DCTN	90.2	92.7	99.4	99.0	95.3
	M3SDA	92.2	94.5	**99.5**	98.2	96.4
	MFSAN	93.8	95.1	99.1	**98.7**	96.7
	MSCLDA	**94.1**	**95.3**	99.1	98.5	**96.8**

weighted combination without adjustment we observed show less difference (the details will be provided below in Fig. 3.8), we only compared results of mean method and weighted method with adjustment. "Domain-level only" means the proposed method without class-level distribution matching, "Multi-level" means the proposed method with domain-level and class-level distributions matching. The source orders are the same as the task name. For example, in task $A, D \rightarrow W$, $S1$ is domain A, $S2$ is domain D. Average result of $S1$ and $S2$ represents average accuracy of all single domain adaptation tasks. "Sbest" means the best performance of single source domain using the proposed method. "MeanC" is the proposed multi-source domain adaptation method using the averaged combination rule.

In most cases, multi-source domain adaptation outperforms single source domain adaptation. Predictions with multi-level distribution matching return higher accuracy than those without matching class-level distribution. Multi-source domain adaptation with weight adjustment outperforms the results without weight adjustment.

For small datasets ImageCLEF-DA and Office-Caltech10, classification accuracy of the proposed method has little difference to that of the average mean method. This may because the performance of single source domain adaptation is fairly similar to other singe source domains. Fig. 3.6 in section 3.4.2 also indicates that the number of the same target labels returned by all single source predictors is near to batchsize, most often being 31 or 32. This represents that here, average mean combination can achieve almost the same performance as weighted average mean combination. It also can be seen that the single best performance of the proposed method is better than single best results provided in Tables 3.2 - 3.5, which means the cross-domain constraint can improve the transferability of single source domain.

To detail the impacts of weights, taking DSLR (target domain) in dataset Office-31 as an example for two-source domain adaptation, and Product in dataset Office-Home as an example for three-source domain adaptation, Fig. 3.8 indicates the results of classification accuracy with and without adjustment.

It can be seen that the accuracy of average mean method and that of weighted mean method without adjustment has no significant difference. The line of average combination is almost superimposed on that of weighted combination without adjustment. If the single source domain adaptation has obvious disparity, the mean combination or weighted combination without adjustment cannot take the advantage of the single best one and return preferable results. The proposed method, however, is superior to them in these cases.

The line of the proposed method displays some fluctuations, that could result

Table 3.6 : Comparison of classification accuracy (%) on dataset Office-31 with different combination rules

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Domain-level only	S1	97.5	97.2	71.3	89.4
	S2	99.6	98.6	72.3	
	Sbest	99.6	98.6	72.3	90.2
	MeanC	99.1	98.3	71.8	89.7
	MSCLDA	99.6	98.7	72.1	90.1
Multi-level	S1	96.7	98.0	72.9	89.9
	S2	99.7	98.7	**73.8**	
	Sbest	99.7	98.7	**73.8**	90.7
	MeanC	98.7	98.7	73.3	90.3
	MSCLDA	**99.8**	**98.8**	73.7	**90.8**

Table 3.7 : Comparison of classification accuracy (%) on dataset ImageCLEF-DA with different combination rules

Standards	Method	I, C→P	I, P→C	P, C→I	Avg
Domain-level only	S1	79.0	95.7	93.1	89.3
	S2	79.0	95.7	93.2	
	Sbest	79.0	95.7	93.2	89.3
	MeanC	79.1	95.8	93.0	89.3
	MSCLDA	79.0	**95.9**	93.2	89.4
Multi-level	S1	79.4	95.7	94.0	89.7
	S2	79.4	95.6	94.2	
	Sbest	79.4	95.7	94.2	89.8
	MeanC	**79.6**	95.7	**94.4**	**89.9**
	MSCLDA	79.5	**95.9**	94.3	**89.9**

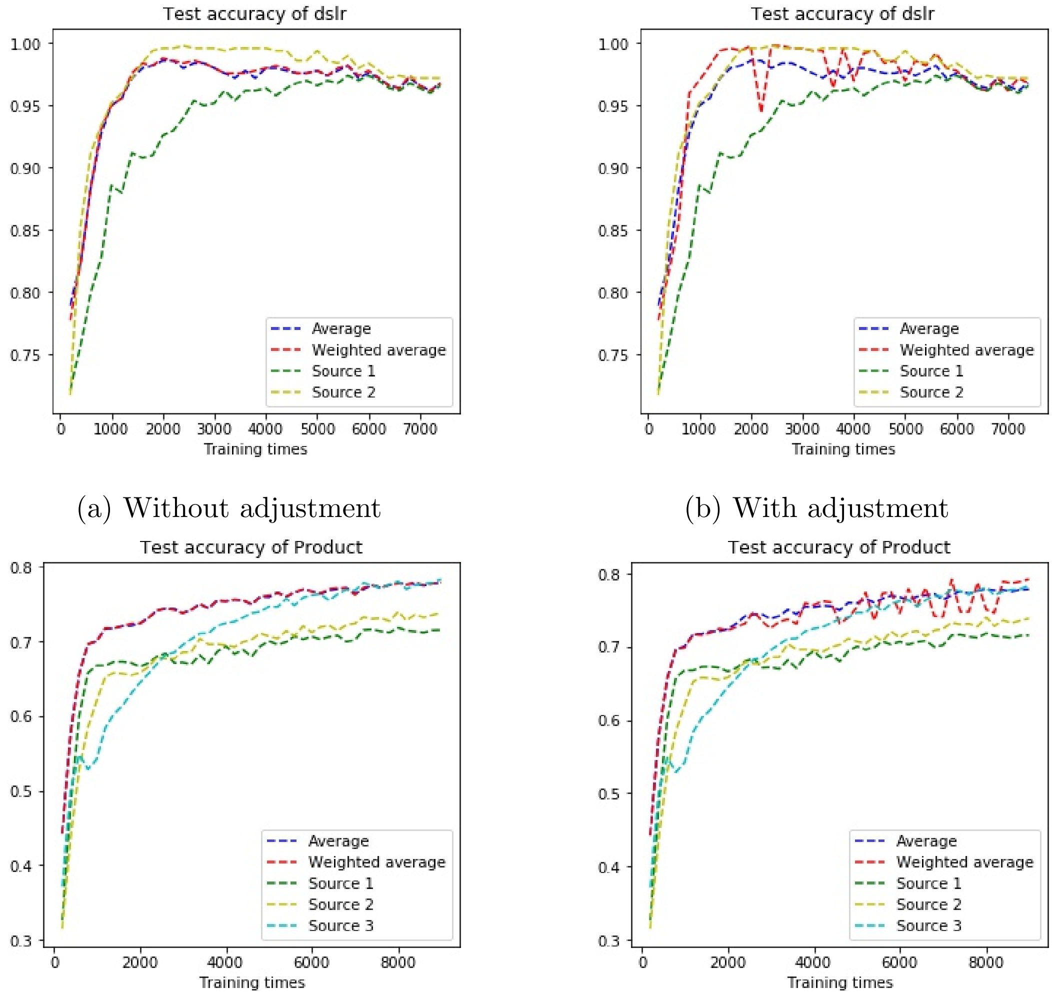
Table 3.8 : Comparison of classification accuracy (%) on dataset Office-Home with different combination rules

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Domain-level only	S1	76.4	72.3	57.6	64.5	68.9
	S2	75.2	73.3	56.7	65.6	
	S3	78.2	78.3	59.4	69.8	
	Sbest	78.2	78.3	59.4	69.8	71.4
	MeanC	80.3	77.8	60.9	69.5	72.1
	MSCLDA	**80.6**	78.8	61.1	70.0	72.6
Multi-level	S1	78.1	73.3	57.8	66.3	71.0
	S2	77.6	77.4	59.5	67.4	
	S3	80.0	**80.3**	61.3	**72.4**	
	Sbest	80.0	**80.3**	61.3	**72.4**	**73.5**
	MeanC	80.4	78.4	61.2	69.7	72.4
	MSCLDA	**80.6**	79.9	**61.4**	71.6	73.4

Table 3.9 : Comparison of classification accuracy (%) on dataset Office-Caltech10 with different combination rules

Standards	Method	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
Domain-level only	S1	92.5	94.8	97.5	96.7	95.8
	S2	92.4	94.4	97.3	98.1	
	S3	93.4	94.3	98.8	99.7	
	Sbest	93.4	94.8	98.8	99.7	96.7
	MeanC	93.7	**95.4**	98.8	98.1	96.5
	MSCLDA	93.7	95.3	98.8	98.5	96.6
Multi-level	S1	93.4	94.8	98.4	96.7	96.2
	S2	93.6	94.4	98.6	96.7	
	S3	**94.1**	94.9	**99.2**	**100.0**	
	Sbest	**94.1**	94.9	**99.2**	**100.0**	**97.1**
	MeanC	**94.1**	**95.4**	99.1	98.5	96.8
	MSCLDA	**94.1**	95.3	99.1	98.5	96.8

from wrong original weights, which means the well performing source domain is given small weight while the inferior ones get large weights. We may take this as future work to explore how to learn more reliable weights that are concordant with their performance on the target domain.



(a) Without adjustment

(b) With adjustment

(c) Without adjustment

(d) With adjustment

Figure 3.8 : Classification accuracy without and with adjusting weights. Figures (a) and (c) are results without adjustment, while figures (b) and (d) are results with adjustment.

Fig. 3.9 shows weights with and without adjustment. Since the adjustment is based on batches, there are too many weights during one experiment. So we randomly choose 50 values of each task to draw the pictures. Taking DSLR (target domain) in dataset Office-31 as an example for two-source domain adaptation, and Product in dataset Office-Home as an examples for three-source domain adaptation, there is some evidence as to why the results of the average mean method is almost the same as the results of weighted mean method without adjustment.

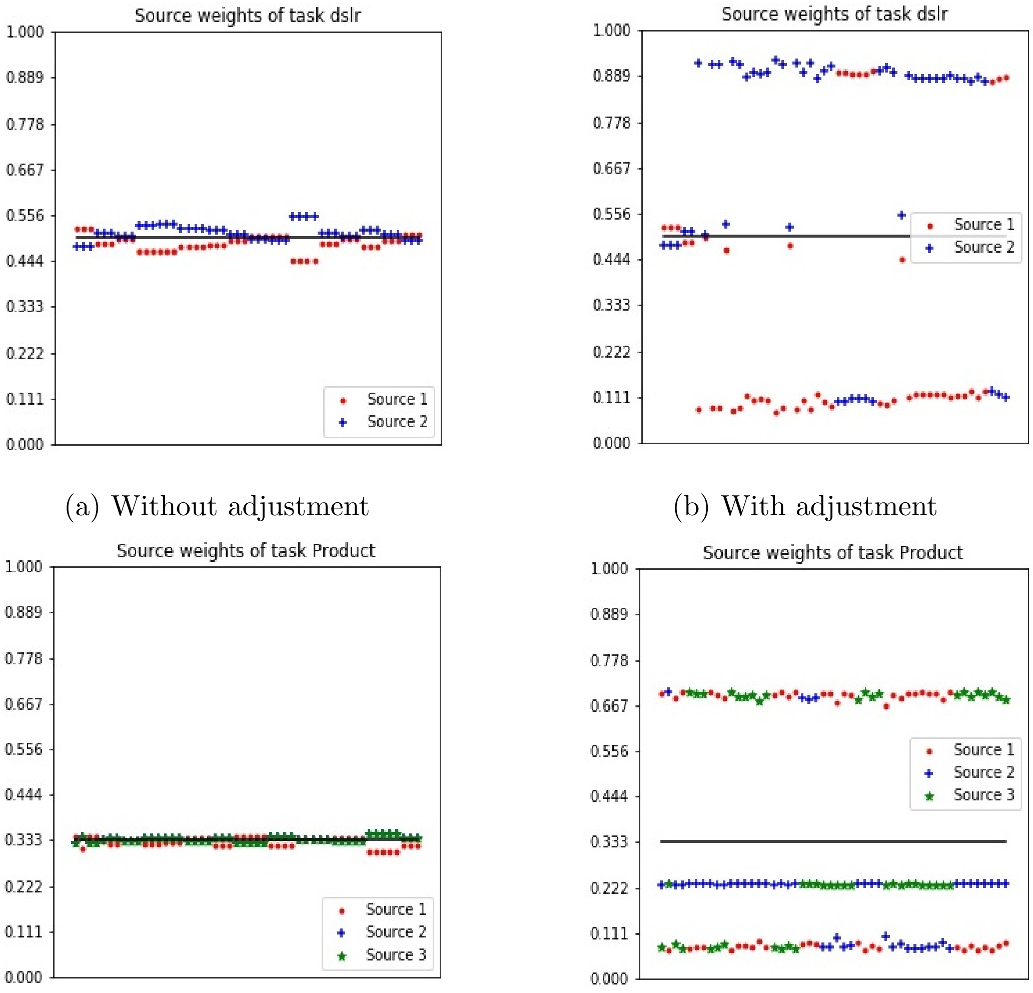
The figure shows that all weights without adjustment are around the mean value of the greater frequencies. Thus, their results rarely differ markedly from each other, while the adjusted weights show significant disparity. For two-source domain adaptation, weights are near to 0 and 1, for three-source domain adaptation, only the smallest weights are near to boundary. It might need further exploration if the largest weights require extra adjustment to make it more closer to 1.

3.4.4 Results and Analysis Based on Fuzzy Rules

Tables 3.10 and 3.11 show the results on ImageCLEF-DA and Office-31 using fuzzy rules respectively.

It can be seen the proposed method achieves the highest performance on most tasks. Generally, multi-source domain adaptation outperforms single source domain adaptation. Knowledge transfer with considering domain shift is superior to which simply mixes all source training samples. Sometimes, single source domain adaptation performs best, for example, tasks $I, C \rightarrow P$ using MDDA and $A, W \rightarrow D$ using DDAN, which means when combining all source classifiers or mixing source samples following different distributions, negative transfer may occur. We will investigate this as future work to avoid negative transfer when combining source domains.

Tables 3.12 and 3.13 show the performance without and with a fuzzy system, “S” means single source domain, “M” means multi-source domain. Source order is the same as described, for example, $S1$ is A in task $A, W \rightarrow D$. It indicates that for many tasks, both single source and multi-source domain adaptation, the



(a) Without adjustment

(b) With adjustment

(c) Without adjustment

(d) With adjustment

Figure 3.9 : Source weights without and with adjusting. Figures (a) and (c) are results without adjustment, while figures (b) and (d) are results with adjustment.

Table 3.10 : Comparison of classification accuracy (%) on dataset ImageCLEF-DA using fuzzy rules

Standards	Method	I, C→P	I, P→C	P, C→I	Avg
Single best	ResNet	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGard	75.0	96.2	87.0	86.1
	MRAN	78.8	95.0	93.5	89.1
	MDDA	**79.8**	95.7	92.0	89.2
	DDAN	78.0	94.0	91.0	87.7
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGard	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN	79.1	95.4	93.6	89.4
	DFRE	79.5	95.8	93.7	89.7
	MDAFuz	79.4	**96.3**	**94.5**	**90.1**

Table 3.11 : Comparison of classification accuracy (%) on dataset Office-31 using fuzzy rules

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Single best	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGard	99.1	96.9	68.2	88.1
	MRAN	99.8	96.9	70.9	89.2
	MDDA	99.2	97.1	73.2	89.8
	DDAN	**100.0**	96.7	65.3	87.3
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGard	99.7	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	DFRE	99.6	98.7	73.1	90.5
	MDAFuz	99.7	**99.0**	**74.0**	**90.9**

performance with fuzzy rules is better than that without fuzzy rules. For some tasks like $A, W \rightarrow D$, the accuracy without fuzzy rules is higher. The reason for this is that source domains show different levels of correlation with the target domain, and for some weakly connected source samples, transferrable information from each cluster is not enough for learning the target task, in other words, the auxiliary among training samples may be lost. We will try to solve this in the future.

Table 3.12 : Comparison of classification accuracy (%) on dataset ImageCLEF-DA without and with fuzzy rules.

Standards		I, C→P	I, P→C	P, C→I	Avg
Without fuzzy	S1	78.8	95.4	93.2	89.2
	S2	79.0	95.2	93.3	
	M	79.1	95.7	93.4	89.4
With fuzzy	S1	78.9	**96.5**	94.3	89.8
	S2	78.7	95.7	**94.8**	
	M	**79.4**	96.3	94.5	**90.1**

3.4.5 Visualization Analysis of Proposed Method

To show the efficiency of domain adaptation using the proposed method, this section displays the visualization of source and target features, which transforms high dimension data into 2-dimension space to display the domain categories directly. Figs. 3.10 and 3.11 show t-SNE visualization (Maaten and Hinton, 2008) of classification features in target domain with different single source domain. Let Office-31 represents two-source domain adaptation and Office-Home represents three-source domain adaptation. For dataset with a small number of categories, it shows that

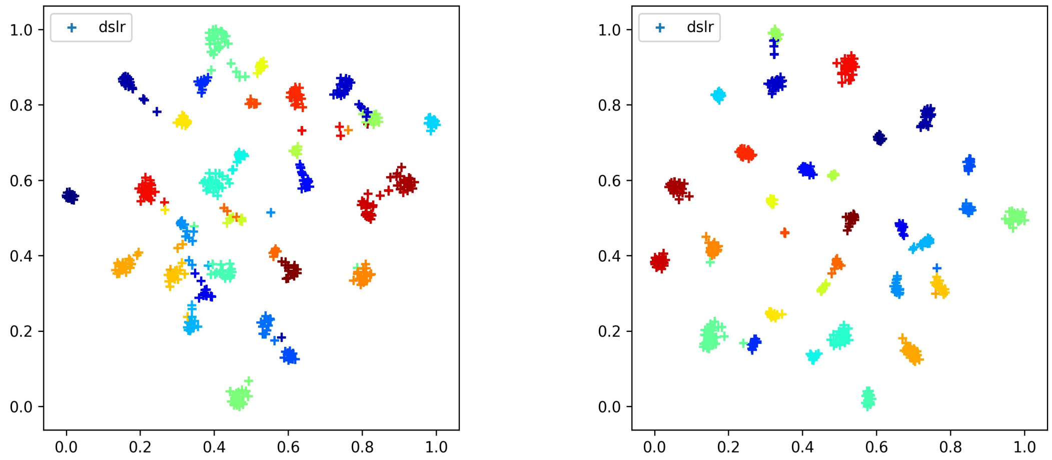
Table 3.13 : Comparison of classification accuracy (%) on dataset Office-31 without and with fuzzy rules.

Standards		A, W-D	A, D-W	W, D-A	Avg
Without	S1	96.3	97.9	73.0	89.8
fuzzy	S2	**99.8**	98.4	73.6	
	M	98.9	98.6	73.3	90.3
With	S1	95.5	98.2	73.0	89.9
fuzzy	S2	99.7	**99.0**	**74.2**	
	M	99.7	**99.0**	74.0	**90.9**

each category separates clearly from others, while task W-D discriminates the categories more clearly than task A-D. This is concordant with classification accuracy shown in Table 3.6, source domain Webcam returns the single best results.

For very large dataset with more categories, only partial boundaries between each two different categories can be discriminated clearly, while the remaining categories may seem too close to each other. Combining results provided in Tables 3.4 and 3.8, the target domain classification accuracy of the whole dataset using different methods falls to a fairly low level compared with other datasets (of which the average accuracy is commonly around 90%). So, it is reasonable that the distance between different categories is not as great as that of datasets with small categories.

To display the effects of distribution matching, Figs. 3.12 and 3.13 show t-SNE visualization of domain features in shared feature space (before adaptation) and multi-view feature space (after adaptation). The red indicates source domain, while blue shows target domain.



(a) A-D

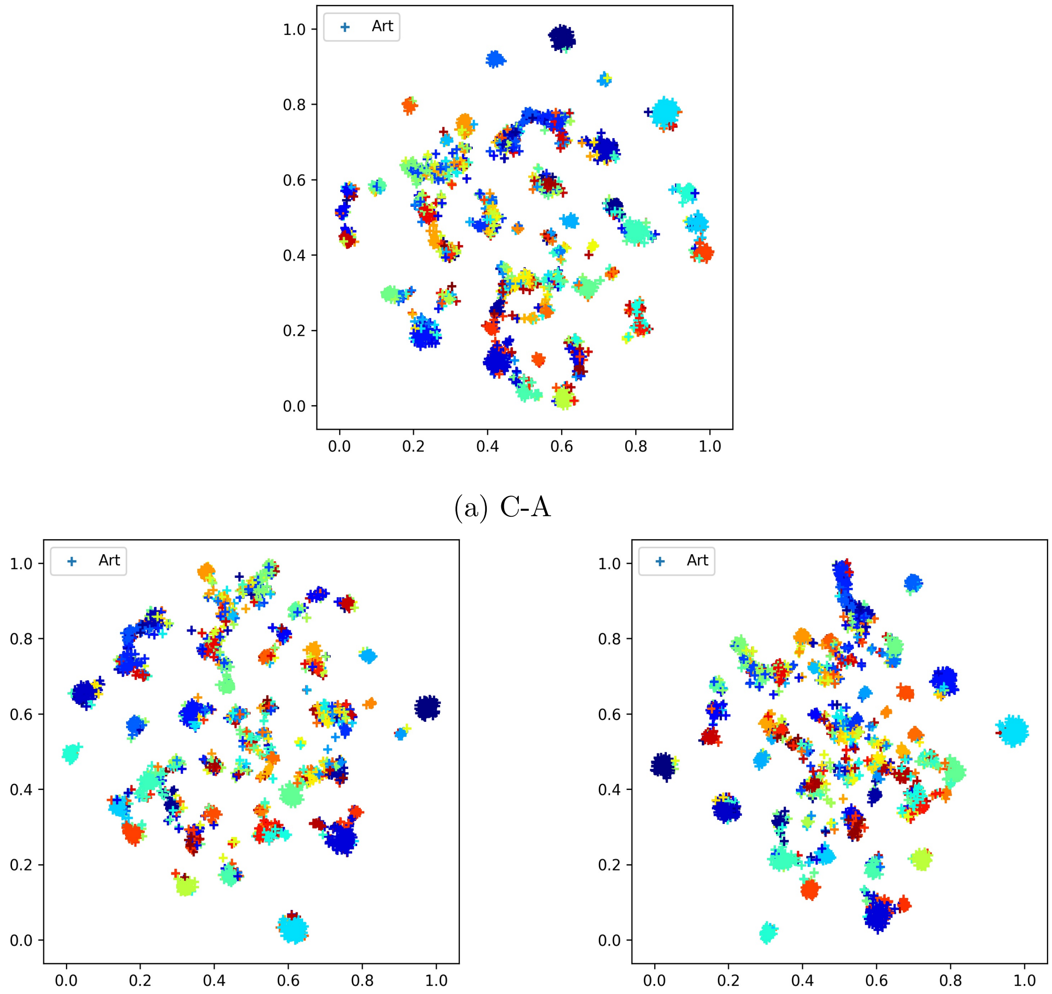
(b) W-D

Figure 3.10 : T-SNE visualization of target with different source domain, take task DSLR in Office-31 as example.

It can be seen that class distribution in multi-view feature space clearly separates from each other while that in shared feature space has misalignments. For all source domains, no matter whether the categories and samples are small or large, the distance between each two classes is sufficient and without any superposition. For target domains, adaptation is achieved well for dataset with small categories, each category in the target domain is mapped with that in the source domain by a short distance. While for Office-Home with large numbers of samples and categories, as mentioned before, the distance among classes after adaptation may not seem as clear as with a small dataset, but, compared with that in shared feature space, the distribution matching still splits the different categories.

3.4.6 Ablation Study and Sample Complexity

An ablation study based on dataset Office-31 is performed to show the effectiveness of the loss components. The constraints of source domain adaptation (L_s), target domain adaptation on domain-level (L_{domain}), target domain adaptation on

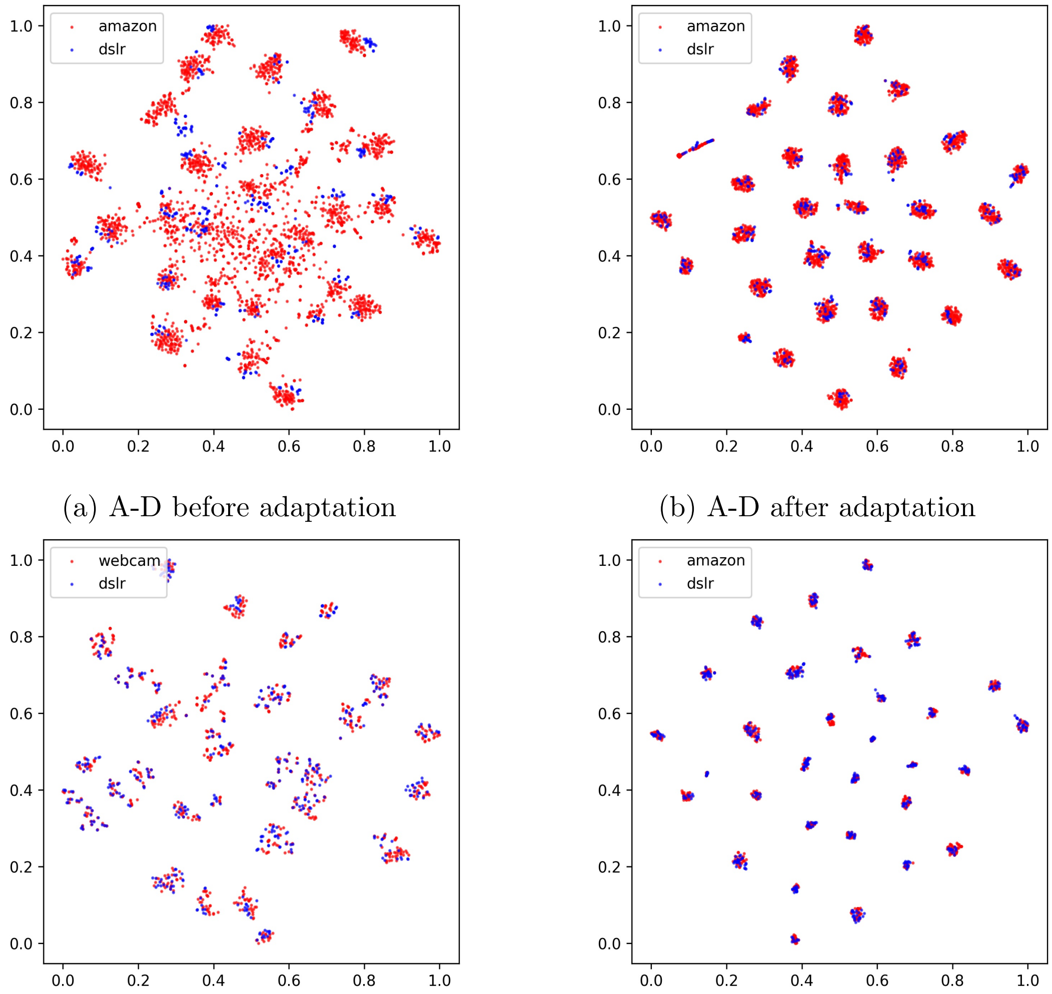


(a) C-A

(b) P-A

(c) R-A

Figure 3.11 : T-SNE visualization of target with different source domain, take task Art in Office-Home as example.



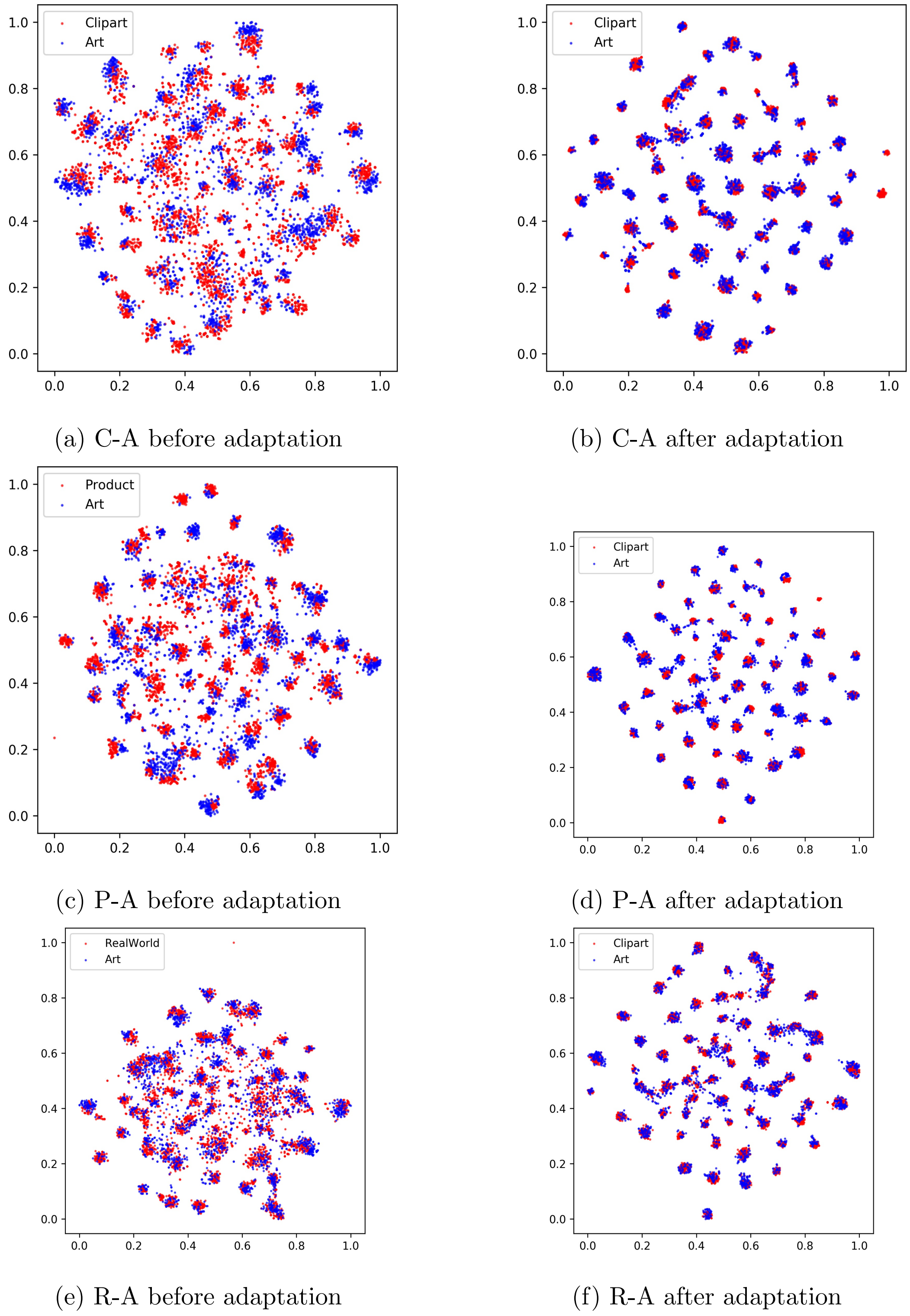
(a) A-D before adaptation

(b) A-D after adaptation

(c) W-D before adaptation

(d) W-D after adaptation

Figure 3.12 : T-SNE visualization of features before and after adaptation, taking task DSLR in Office-31 as example for two-source domain adaptation.



(a) C-A before adaptation

(b) C-A after adaptation

(c) P-A before adaptation

(d) P-A after adaptation

(e) R-A before adaptation

(f) R-A after adaptation

Figure 3.13 : T-SNE visualization of features before and after adaptation, taking task Art in Office-Home as example for three-source domain adaptation.

class-level (L_{class}) and cross-domain alignment (L_{cro}) are regarded as the control variables, and each of them is removed in turn to show its contribution of the learning performance. All experiments are repeated for three times, and the results are shown in Table. 3.14.

It can be seen that the target domain adaptation on domain-level and the cross-domain alignment contribute more than the other constraints, because the classifier trained without either of them returns the lowest accuracy. Domain adaptation on class-level plays an auxiliary role of domain-level adaptation to help improve the performance. Source domain adaptation shows superiority in the target task which contains a large number of samples.

Table 3.14 : Accuracy (%) without different loss components

Standards	A, W→D	A, D→W	W, D→A	Avg
Without L_s	**99.9**	**98.9**	73.0	90.6
Without L_{domain}	99.7	98.1	71.4	89.7
Without L_{class}	99.7	**98.9**	72.5	90.4
Without L_{cro}	99.8	97.9	71.0	89.6
Proposed	99.8	**98.9**	**73.9**	**90.9**

To show the influence of training sample size on the learning performance, sample complexity experiments are taken on dataset Office-31. For each task, we randomly select 25%, 50% and 75% source samples to train the classifier and compare its performance with the classifier that is trained using all source samples. The results are shown in Table 3.15.

It can be seen that with the growth of the training sample size, the performance improves. The greatest increase occurs when the sample size increases from 25% to

50%, after which the growth slows. For task $A, D \rightarrow W$, the performance of 50% and 75% samples are extremely close to each other. This indicates that when the pre-trained networks are used as the backbone, the training sample quantity might not be the only main factor affecting the learning performance. Other factors such as the sample quality and domain similarity should also be taken into consideration.

Table 3.15 : Accuracy (%) with different training sample size.

Standards	A, W→D	A, D→W	W, D→A	Avg
25% samples	95.6	96.3	67.1	86.3
50% samples	97.5	98.2	71.7	89.1
75% samples	98.7	98.1	73.4	90.1
100% samples	**99.8**	**98.9**	**73.9**	**90.9**

3.5 Summary

This section concludes the whole work and formulates the directions for further study.

In this chapter, we propose a source contributions learning method for multi-source domain adaption, where the multi-view feature extraction and multi-level distribution matching are employed to enhance transferability of domain adaptation. Compared with existing multi-source domain adaptation methods, ours not only explores the similarities among source and target domains, but also learns diversities of a target domain and turns it into extracting multiple aspects of source domain features since the target data is unlabeled. At the same time, domain adaptation is achieved by adapting source domains to each other as well as adapting source and target domains using domain-level distribution matching and class-level distribution matching, which improve the classification accuracy by reducing the confusion of boundary samples. When it comes to completing a target task, weight adjustment strategy and fuzzy combination rule are developed based on pseudo target labels to increase the disparity of source weights, which can take advantage of the single best source domain when the performances of sources have significant differences. Experiments on real-world visual datasets evaluate the superiority of the proposed method compared with other state-of-the-art domain adaptation methods using deep neural networks.

In the future, we might explore new methods to learn more reliable weights that can represent their performance on target domain exactly. By doing this, we expect to solve the problem where the source domain with poor accuracy on target attracts large weight. Another work is extending the proposed method to heterogeneous feature spaces. Features from different views may have their own best represented ability with different dimensions. This still needs further exploration.

Chapter 4

Dynamic Classifier Alignment for Unsupervised Multi-Source Domain Adaptation

4.1 Introduction

As introduced in Chapter 2, a popular way to overcome the data bias in transfer learning is to find a mapping that transforms the source and target data into a latent feature space where their distributions can be matched. Some methods employ multi-level matching to minimize the distance of source and target distributions (Tian et al., 2020a), such as pixel-level and feature-level (Xu et al., 2020a), classification and clustering distribution adaptations (Pan et al., 2020).

To explore richer usable information from the target domain, pseudo labels are employed to fill the domain gap by self-training the transfer model (Chen et al., 2019). A typical method, self-supervised noisy label learning, addresses source-free domain adaptation by transforming the problem into label denoising (Chen et al., 2021). It divides the target data into a clean part and a noisy part based on the loss of pre-generating pseudo labels using the source-only classifier, at the same time, self-generated target labels are collected using k-means clustering. By jointly training the clean part with pre-generated labels and the noisy part with self-generated labels, true-labeled samples in the noisy set will be selected into the clean set until the classifier is fixed with the smallest loss.

How to collect enough information for transferring is essential when leveraging source knowledge to solve the target task. Domain adaptation methods with multi-view representations and multiple sources attract considerable attention since they can provide richer transferable knowledge (Niu et al., 2015; Lu et al., 2020). Multi-view features enrich the usable information on the data-level, while multiple sources enrich domain-level knowledge. Multi-view learning for domain adaptation first unifies the problems of view alignment and knowledge transfer (Ding et al., 2018b). To fuse view-invariant knowledge from multi-viewpoint features, mappings among multiple views are introduced to learn the data correspondence in a common space. To transfer knowledge to the target view, marginal and conditional distributions are adapted simultaneously in both sample space and feature space. To avoid the incompleteness of categories in a single source domain, multiple source domains are explored to develop a cross-domain and cross-source algorithm which overcomes the gaps resulting from missing categories.

However, many previous domain adaptation methods which employ multi-view features focus on merging features in a common space or simply concatenate them together but ignore the view-specific information or the importance of each view of features. Additionally, some methods mix domains to learn a general performed classifier on multiple domains but disregard the fact that a common feature space for all domains may not exist (Ben-David et al., 2010). In this chapter, we propose an unsupervised multi-source domain adaptation method named dynamic classifier alignment (DCA), which explores the importance of multi-view features and aligns the multiple view predictions from multiple domains via an automatic method to complete the target task. Our contributions are threefold:

- We propose a new method to learn the importance of multi-view features and re-weight them to ensure the dominant features contribute more when merging their predictions. Different from existing methods which treat all view features equally and connect them as a series, the proposed method considers specific information carried by features and avoids information redundancy resulting from concatenation.

- We build a self-training strategy by selecting pseudo target labels with high confidence in the training progress. It improves the cross-domain ability of the source classifiers by iteratively splitting the target domain into training and testing sets. Different from many pseudo-labeling domain adaptation methods, no prototypes are needed to initialize the labels.

- We develop an automatic sample-wise method to learn the weight vectors for conjoining multiple predictions from different views and source classifiers to estimate the target labels. This differs from existing combination rules relying on feature distance. Meanwhile, the proposed method explores the view importance and the correlation between domains over the sample not over the batch, which is more accurate.

4.2 Problem Setting and Notations

We focus on homogeneous unsupervised multi-source domain adaptation. Table 4.1 displays the notations and corresponding descriptions used in this chapter.

Table 4.1 : Notations and descriptions.

Notation	Description
$\mathcal{D}_{s_k}$, $\mathcal{D}_t$	source/target domain, k is source index
n_{s_k}, n_t	number of samples from source/target domain
$\boldsymbol{x}_{s_k}$, $\boldsymbol{x}_t$	sample from the source/target domain
$\boldsymbol{y}_{s_k}$	corresponding label of $\boldsymbol{x}_{s_k}$
ϕ	shared feature extractor
ϕ_{km}	mth view feature extractor for kth source domain
M_k	number of views for kth source domain
$\boldsymbol{f}_{s_k}$, $\boldsymbol{f}_t$	shared source / target feature
$\boldsymbol{f}_{s_{km}}$, $\boldsymbol{f}_{t_{km}}$	mth view specific source/target feature
$P_{s_{km}}$	mth view classifier for kth source domain
$\boldsymbol{\theta}_{s_{km}}$	parameter of $P_{s_{km}}$ transforming input to output
P_k^a	auxiliary classifier for kth source domain
G_k^c	importance learning function
P_k^c	aligned classifier for kth source domain
$\boldsymbol{\omega}_k^a$	generation parameter for kth source domain
$\boldsymbol{\omega}_k^c$	alignment parameter for kth source domain
G^d	domain discriminator
$\boldsymbol{\omega}$	domain combination parameter
P_t	target classifier
$\mathcal{H}$	reproducing kernel Hillbert space (RKHS)
ψ	kernel-induced feature map

4.3 The Proposed Dynamic Classifier Alignment

We focus on homogeneous unsupervised multi-source domain adaptation. The whole framework of the proposed classifier alignment method is shown in Fig. 4.1. It includes feature extraction, classifier alignment, pseudo label selection and target task completion. Feature extraction collects shared features and specific multi-view features. Shared source features are used to learn a domain discriminator which identifies the probabilities of a sample belonging to the source domains, while adapted specific multi-view features are used to train a series of classifiers. An auxiliary classifier of each source domain is generated by linearly mixing the multiple classifiers learned from different views, then an importance learning function is developed based on the auxiliary classifier to learn the importance of multi-view features for aligning classifiers. At the same time, pseudo labels are provided by the learned classifiers to supervise further training. Using the degrees of the target samples belonging to the source domains calculated by the domain discriminator trained with shared features, source classifiers are finally combined to predict the target labels.

4.3.1 Feature Extraction

In an image classification task, original images are transformed into a feature space to avoid its original high dimension and complex structure. Since invariant information from the source and target domains is needed to achieve transfer, to measure the similarity between domains, pre-trained deep networks are first employed to transform the original data to extract those features carrying invariant information. We note the features extracted using pre-trained networks as shared features, where the term "shared" indicates the extracted features are in a same latent space. The extraction can be expressed as:

$$\begin{aligned} \boldsymbol{f}_{s_k} =& \phi(\boldsymbol{x}_{s_k}), \\ \boldsymbol{f}_t =& \phi(\boldsymbol{x}_t). \end{aligned} \tag{4.1}$$

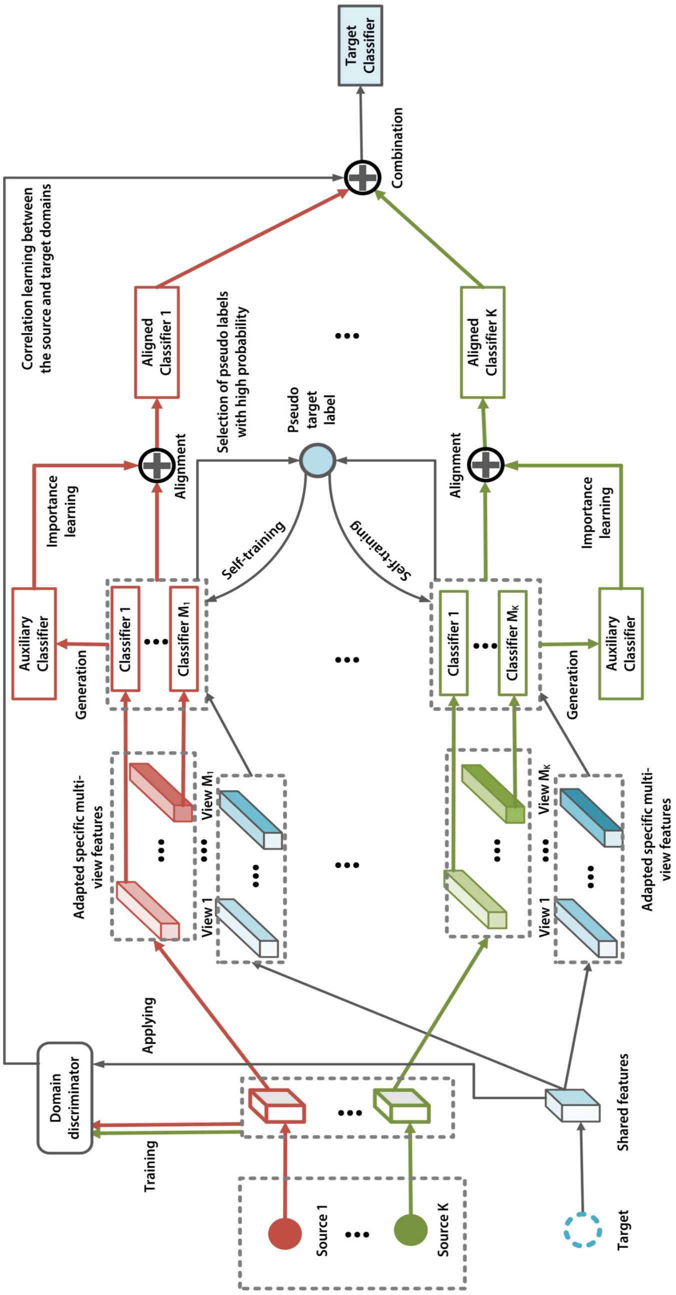


Figure 4.1 : The whole framework of the proposed DCA. Shared features are extracted to learn a domain discriminator which defines the contributions of source domains when predicting target task. Specific multi-view features are extracted to adapt domain distributions and train classifiers. Auxiliary classifier in each source domain is generated to assist in aligning multi-view classifiers. Target samples are pseudo-labeled by the learned classifiers to supervise the further training.

Shared features can carry invariant information from source and target domains. Based on this, a domain discriminator driven from source features is trained, which calculates the probabilities of extracted shared features belonging to the source domains. The correlation between the source and target domains can be obtained to assist in predicting the target task when applying the learned domain discriminator to target features. The domain discriminator G^d can be expressed as:

$$\begin{aligned} G^d =& \arg\min_{G^d} \; L(G^d(\boldsymbol{f}_{s_k}), d) \\ =& \arg\min_{G^d} \; -\frac{1}{\sum_{k=1}^{K} n_{s_k}} \sum_{i=1}^{\sum_{k=1}^{K} n_{s_k}} d^i \log(G^d(\boldsymbol{f}_{s_k}^i)) \end{aligned} \tag{4.2}$$

L is cross-entropy loss, d indicates domain label.

As proved in (Ben-David et al., 2010), a feature space that can represent all domains well may not exist, hence specific features are needed to adapt each pair of source and target domains. Here we extract multi-view specific features. For the kth source domain, the mth view features can be expressed as:

$$\begin{aligned} \boldsymbol{f}_{s_{km}} =& \phi_{km}(\boldsymbol{f}_{s_k}), \\ \boldsymbol{f}_{t_{km}} =& \phi_{km}(\boldsymbol{f}_t). \end{aligned} \tag{4.3}$$

To provide the cross-domain ability of the source classifiers, source and target distributions are matched to reduce the discrepancy. Here MMD is employed to measure the distribution distance. According to previous research (Pan et al., 2010), given data sets $\boldsymbol{X}_\star, \boldsymbol{X}_*$ following different distributions, the solution of MMD can be written as:

$$\begin{aligned} & \mathcal{MMD}_{\psi \in \mathcal{H}}(\boldsymbol{X}_\star, \boldsymbol{X}_*) \\ & \quad = \left\| \frac{1}{n_\star} \sum_{i=1}^{n_\star} \psi(\boldsymbol{X}_\star^i) - \frac{1}{n_*} \sum_{j=1}^{n_*} \psi(\boldsymbol{X}_*^j) \right\|_{\mathcal{H}}^2 , \end{aligned} \tag{4.4}$$

where $n_\star, n_*$ are corresponding sample quantities, and $\mathcal{H}$ represents reproducing kernel Hilbert space, ψ is kernel-induced feature map. Applying this to the extracted multi-view specific features, for the kth source domain, we adapt each view of the source and target features to ensure that each view source classifier has the ability to predict target samples transformed into the same view feature space. The loss function of domain-level adaptation on mth view features can be expressed as:

$$\begin{aligned} L_{adpt} =& \underset{\mathcal{K}\in\mathcal{H}}{\mathcal{MMD}}(\boldsymbol{f}_{s_{km}}, \boldsymbol{f}_{t_{km}}) \\ =& \left\| \frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \psi(\boldsymbol{f}^i_{s_{km}}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \psi(\boldsymbol{f}^j_{t_{km}}) \right\|^2_{\mathcal{H}} . \end{aligned} \tag{4.5}$$

During the training, since the parameters are updated over the batch, n_{s_k}, n_t can be replaced with batch size.

4.3.2 Classifier Alignment

Generally, aligning multi-view features is more common to learn a classifier. However, concatenating features may result in redundant information. Some studies mix features, but features from different views may have different dimensions, causing a heterogeneity problem. This chapter proposes classifier alignment. In terms of the structural risk minimization principle (Vapnik and Vapnik, 1998), the loss of supervised learning processing of the kth source classifier based on the mth view features $P_{s_{km}}$ can be expressed as:

$$P_{s_{km}} = \underset{\substack{P_{s_{km}}\in\mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k})\sim\mathcal{D}_{s_k}}}{\arg\min} \; L(P_{s_{km}}(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) + \lambda L_{adpt}. \tag{4.6}$$

L is the cross-entropy loss estimating the error between the predictions and the ground truth labels, which is:

$$\begin{aligned} & L(P_{s_{km}}(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) \\ =& -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}^i_{s_k} \log(P_{s_{km}}(\phi_{km}(\phi(\boldsymbol{x}^i_{s_k})))). \end{aligned} \tag{4.7}$$

To define the alignment parameters, an auxiliary classifier for each source domain is generated first. It can expressed as:

$$P^a_k = G^a_k((\{P_{s_{km}}\}^{M_k}_{m=1}); \boldsymbol{\omega}^a_k) = \sum_{m=1}^{M_k} \omega^a_{km} \cdot P_{s_{km}}, \tag{4.8}$$

where M_k is number of views, $\boldsymbol{\omega}_k^a = [\omega_{k1}^a, \cdots, \omega_{kM_k}^a]$ is initialized randomly, G_k^a is a linear function. However, this combination cannot ensure that the weight ω_{km}^a is nonnegative or $\sum_{m=1}^{M_k} \omega_{km}^a = 1$. In addition, ω_k^a is shared by all samples without considering the characteristic of each sample. Hence, we build an importance learning function based on the the auxiliary classifier to fix this problem, which calculates the weights dynamically over the sample. This function estimates the contributions of multi-view classifiers, and the constraint of the function in the kth source domain is:

$$\begin{aligned} G_k^c =& \arg\min_{G_k^c} \sum_{m=1}^{M_k} L(G_k^c(P_{s_{km}}(\boldsymbol{x}_{s_k})), m) \\ =& \arg\min_{G_k^c} \sum_{m=1}^{M_k} (-\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} m^i \log(G_k^c(P_{s_{km}}(\boldsymbol{x}_{s_k}^i)))), \end{aligned} \tag{4.9}$$

m is the view label. Apply the auxiliary classifier to source samples and feed corresponding outputs to the importance learning function, the weight vector is:

$$\boldsymbol{\omega}_k^c = G_k^c(P_k^a(\boldsymbol{x}_{s_k})) = [\omega_{k1}^c, \cdots, \omega_{kM_k}^c]. \tag{4.10}$$

The aligned classifier is:

$$P_k^c = \sum_{m=1}^{M_k} \omega_{km}^c \cdot P_{s_{km}}. \tag{4.11}$$

Importance learning is further controlled by minimizing the error between the auxiliary classifier and the aligned classifier under the supervision of the source labels. Loss function of the matching classifiers can be expressed as:

$$\begin{aligned} L_{ca} =& L_1(P_k^c, P_k^a) \\ =& \frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \left| P_k^c(\boldsymbol{x}_{s_k}^i) - P_k^a(\boldsymbol{x}_{s_k}^i) \right| \\ =& \frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \Big| \sum_{m=1}^{M_k} \omega_{km}^c \cdot P_{s_{km}}\big(\phi_{km}(\phi(\boldsymbol{x}_{s_k}^i))\big) \\ & - \sum_{m=1}^{M_k} \omega_{km}^a \cdot P_{s_{km}}\big(\phi_{km}(\phi(\boldsymbol{x}_{s_k}^i))\big) \Big|. \end{aligned} \tag{4.12}$$

The supervision loss is:

$$\begin{aligned} L(P_k^c(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) = \\ -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(\sum_{m=1}^{M_k} \omega_{km}^c \cdot P_{s_{km}}(\phi_{km}(\phi(\boldsymbol{x}_{s_k}^i)))). \end{aligned} \tag{4.13}$$

Since we have multiple source domains, it is expected that the multiple source classifiers can obtain the same labels when predicting the same target samples. Hence, cross-domain constraint is employed to reduce the predicted target errors between multiple classifiers from different source domains, which can be expressed as:

$$\begin{aligned} L_{cro} = & \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\ & (\frac{1}{n_t} \sum_{i=1}^{n_t} \left| P_{k_1}^c(\boldsymbol{x}_t^i) - P_{k_2}^c(\boldsymbol{x}_t^i) \right|) \\ = & \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\ & (\frac{1}{n_t} \sum_{i=1}^{n_t} \Big| \sum_{m=1}^{M_{k_1}} \omega_{k_1 m}^c \cdot P_{s_{k_1 m}}\big(\phi_{k_1 m}(\phi(\boldsymbol{x}_t^i))\big) \\ & - \sum_{m=1}^{M_{k_2}} \omega_{k_2 m}^c \cdot P_{s_{k_2 m}}\big(\phi_{k_2 m}(\phi(\boldsymbol{x}_t^i))\big) \Big|). \end{aligned} \tag{4.14}$$

Then the total loss function of the aligned source classifier is:

$$\begin{aligned} L_{total} = & L(P_k^c(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) \\ & + \sum_{m=1}^{M_k} (L(P_{s_{km}}(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) + \lambda L_{adpt}) \\ & + \alpha \sum_{m=1}^{M_k} L(G_k^c(P_k^a(\boldsymbol{x}_{s_k})), m) + \beta L_{ca} + \gamma L_{cro}. \end{aligned} \tag{4.15}$$

4.3.3 Pseudo Label Selection and Target Task Completion

To enhance the cross-domain ability of the learned source classifiers, an extra constraint -the supervision of pseudo labels -is introduced to improve the transfer performance from the source domains to the target domain except for the cross-domain constraint.

Not all pseudo labels can be used to supervise the training because of the label noise. It is very risky to introduce incorrect target labels which may degrade the performance. To collect the pseudo labels with high a probability of being correct, a selection strategy is developed, thresholds E and step q relating to the training progress are set to define when the pseudo labels can be added. As shown in Fig. 4.2, unlabeled target samples are fed into the source classifiers from multiple views, and the outputs can be divided into two groups, namely easy-to-predict and hard-to-predict samples. Easy-to-predict samples obtain the same labels with high probability, while hard-to-predict samples obtain multiple labels or obtain the same labels with low probability. Pseudo-labeled target samples with high confidence are collected for further training.

Since we have multiple view classifiers in each source domain, when the training iteration is larger than E, denote the pseudo target label with a high confidence of being correct and the corresponding probability as:

$$\begin{aligned}\hat{\boldsymbol{y}}_t = \wedge(&P_{s_{11}}(\phi_{11}(\phi(\boldsymbol{x}_t))), \cdots, P_{s_{1M_1}}(\phi_{1M_1}(\phi(\boldsymbol{x}_t))),\\ &P_{s_{21}}(\phi_{21}(\phi(\boldsymbol{x}_t))), \cdots, P_{s_{2M_2}}(\phi_{2M_2}(\phi(\boldsymbol{x}_t))),\\ &\cdots,\\ &P_{s_{K1}}(\phi_{K1}(\phi(\boldsymbol{x}_t))), \cdots, P_{s_{KM_k}}(\phi_{KM_k}(\phi(\boldsymbol{x}_t)))),\end{aligned} \tag{4.16}$$

and

$$\begin{aligned}\hat{p}_t =&\frac{1}{\sum_{k=1}^{K} M_k} \max\\ &(P_{s_{11}}(\phi_{11}(\phi(\boldsymbol{x}_t))) + \cdots + P_{s_{1M_1}}(\phi_{1M_1}(\phi(\boldsymbol{x}_t)))+\\ &P_{s_{21}}(\phi_{21}(\phi(\boldsymbol{x}_t))) + \cdots + P_{s_{2M_2}}(\phi_{2M_2}(\phi(\boldsymbol{x}_t)))+\\ &\cdots+\\ &P_{s_{K1}}(\phi_{K1}(\phi(\boldsymbol{x}_t))) + \cdots + P_{s_{KM_k}}(\phi_{KM_k}(\phi(\boldsymbol{x}_t)))),\end{aligned} \tag{4.17}$$

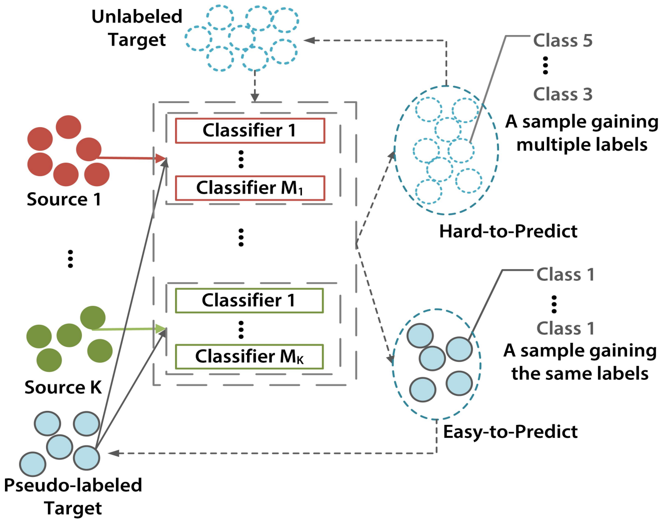


Figure 4.2 : The procedure of selecting pseudo labels. Solid arrows indicate training under the supervision of source labels and target pseudo labels. Dashed arrows indicate the prediction and selection of target samples. At the beginning, classifiers are supervised by source labels only. Unlabeled target samples are fed into the learned classifiers to be divided into easy-to-predict and hard-to-predict groups. Predicted target labels with high probability are selected to supervise further training of the classifiers.

respectively. $\wedge$ is an operation to select the target samples which obtain the same predicted labels using multiple classifiers. A threshold a^r for the rth category is defined to select pseudo labels for training. If $\hat{p}_t \geq a^r$, we regard the corresponding target sample as a high confident sample. Otherwise, it will not be selected. The collected pseudo target samples will be used to supervise the further training. The selection is taken iteratively (by step q) until the training ends or there are no unlabeled target samples available. The loss function of the kth source classifier under pseudo label supervision can be expressed as:

$$\begin{aligned} L_t =& \sum_{m=1}^{M_k} L(P_{s_{km}}(\boldsymbol{x}_t), \hat{\boldsymbol{y}}_t) + L(P_k^c(\boldsymbol{x}_t), \hat{\boldsymbol{y}}_t) \\ =& -\sum_{m=1}^{M_k} (\frac{1}{\hat{n}_t} \sum_{i=1}^{\hat{n}_t} \hat{\boldsymbol{y}}_t^i \log(P_{s_{km}}(\phi_{km}(\phi(\boldsymbol{x}_t^i)))) \\ & -\frac{1}{\hat{n}_t} \sum_{i=1}^{\hat{n}_t} \hat{\boldsymbol{y}}_t^i \log(\sum_{m=1}^{M_k} \omega_{km}^c \cdot P_{s_{km}}(\phi_{km}(\phi(\boldsymbol{x}_t^i)))), \end{aligned} \tag{4.18}$$

The loss function in (4.15) of further training can be rewritten as following when pseudo labels are available:

$$\begin{aligned} L_{total} =& L(P_k^c(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) \\ &+ \sum_{m=1}^{M_k} (L(P_{s_{km}}(\boldsymbol{x}_{s_k}), \boldsymbol{y}_{s_k}) + \lambda L_{adpt}) \\ &+ \alpha \sum_{m=1}^{M_k} L(G_k^c(P_k^a(\boldsymbol{x}_{s_k}), m) + \beta L_{ca} + \gamma L_{cro} + L_t. \end{aligned} \tag{4.19}$$

To complete the target task, combining source predictions is a common strategy. Employing the domain discriminator learned in equation (4.2), then the correlation between a target sample and the source domains can be calculated as:

$$\boldsymbol{\omega} = G^d(\boldsymbol{f}_t) = [\omega_1, \cdots, \omega_K]. \tag{4.20}$$

Target classifier can be expressed as:

$$P_t = \sum_{k=1}^{K} \omega_k \cdot P_k^c. \tag{4.21}$$

The target label is:

$$\begin{aligned}\boldsymbol{y}_t =& P_t(\boldsymbol{x}_t)\\ =& \sum_{k=1}^{K} \omega_k \cdot P_k^c(\boldsymbol{x}_t)\\ =& \sum_{k=1}^{K} \omega_k \cdot (\sum_{m=1}^{M_k} \omega_{km}^c \cdot P_{s_{km}}(\boldsymbol{x}_t)).\end{aligned} \tag{4.22}$$

The process of the proposed classifier alignment is described in Algorithm 3.

Algorithm 3 Dynamic classifier alignment for multi-source domain adaptation

```
Input: Source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$, target domain $\mathcal{D}_t$;
Initialization: Shared feature extraction network $\phi$, multi-view specific feature extraction networks $\{\phi_{km}\}_{k,m=1}^{K,M_k}$ and multi-view classifiers $\{P_{s_{km}}\}_{k,m=1}^{K,M_k}$, auxiliary classifiers $\{P_k^a\}_{k=1}^{K}$, discriminators $\{G_k^c\}_{k=1}^{K}$ and $G^d$.
for $e = 1$, $e < \mathcal{I}$, $e++$, do
    Extract shared features $\boldsymbol{f}_{s_k}$, $\boldsymbol{f}_t$ as in (4.1);
    Train domain discriminator as in (4.2);
    Extract multi-view specific features $\boldsymbol{f}_{s_{km}}$, $\boldsymbol{f}_{t_{km}}$ as in (4.3);
    Calculate adaptation loss as in (4.5);
    Train classifiers as in (4.6);
    Align source classifiers as in (4.11);
    if pseudo label available then
        Collect pseudo target label and corresponding probability $\hat{\boldsymbol{y}}_t$, $\hat{p}_t$ for further training as in (4.16), (4.17);
        Select pseudo target labels according to the threshold $a^r$;
        Update all parameters as in (4.19);
    else
        Update all parameters as in (4.15);
    end if
end for
Predict target labels as in (4.22);
Output: Target label $\boldsymbol{y}_t$.
```

4.4 Experiments

This section discusses the results of a series of experiments on four commonly used real-world visual datasets. The classification performance, parameter sensitivity and an ablation study of the proposed method are explored. Section 4.4.1 introduces the datasets and baselines. Section 4.4.2 details the parameter settings and the influence of adjusting the learning rate when introducing pseudo labels. Section 4.4.3 analyses the results of the proposed method and the baselines. Section 4.4.4 compares the performance between a single-view classifier and an aligned classifier. Section 4.4.5 analyses the results of using feature alignment and classifier alignment. Section 4.4.6 explores when to add pseudo labels, Section 4.4.7 describes an ablation study of the proposed method and Section 4.4.8 shows data visualization in multiple views.

4.4.1 Baselines and Datasets

All our experiments focus on the image classification task, baselines contains both single source and multi-source domain adaptation methods. Single source methods include:

- Deep adaptation network (DAN) (Long et al., 2015);
- Reverse gradient (RevGrad) (Ganin and Lempitsky, 2015);
- Correlation alignment for domain adaptation (D-CORAL) (Sun and Saenko, 2016);
- Multi-representation adaptation network (MRAN) (Zhu et al., 2019b);
- Multi-adversarial domain adaptation (MADA) (Pei et al., 2018);
- Manifold dynamic distribution adaptation (MDDA) (Wang et al., 2020b);

- Dynamic distribution adaptation network (DDAN) (Wang et al., 2020b);
- Adversarial discriminative domain adaptation (ADDA) (Tzeng et al., 2017);
- Cycle-consistent adversarial domain adaptation (CyCADA) (Hoffman et al., 2018);
- Adversarial-learned loss for domain adaptation (ALDA) (Chen et al., 2020).

Multi-source methods include:

- Deep cocktail network (DCTN) (Xu et al., 2018);
- Moment matching for multi-source domain adaptation (M3SDA) (Peng et al., 2019a);
- Multiple feature spaces adaptation network (MFSAN) (Zhu et al., 2019a);
- Multi-source distilling domain adaptation (MDAN) (Zhao et al., 2020a);
- Multi-source adversarial domain aggregation network (MADAN) (Zhao et al., 2021);
- Online meta-learning for multi-source domain adaptation (MetaMDA) (Li and Hospedales, 2020);
- Multi-source contribution learning for domain adaptation (MSCLDA) (Li et al., 2021c).

The experiment datasets are ImageCLEF-DA, Office31, Office-Caltech10 and OfficeHome. Office-31 contains 4110 images from three libraries sharing 31 categories in total. The images are captured by different photographic devices, named Amazon (A), Webcam (W) and DSLR (D). Amazon comprises 2817 images, Webcam comprises 795 and DSLR comprises 498.

ImageCLEF-DA includes three image libraries and contains 1800 images. It is built by collecting the 12 shared categories from the datasets Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P), and each categories holds 50 images and every library holds 600 images in total.

Office-Caltech10 consists of four libraries with 10 categories shared by datasets Office-31 and Caltech-256. It has 2533 images in total, where the library Caltech (C) holds 1123 images, Amazon (A) holds 958 images, Webcam (W) holds 295 images, and DSLR (D) holds 157 images.

Office-Home contains 15588 images sharing 65 categories. It has four image libraries named datasets Art (A), Clipart (C), Product (P) and Real World (R). Library Art holds 2427 images, Clipart holds 4365 images, Product holds 4439 images, and Real World holds 4357 images.

Let one of the libraries in each dataset be the target domain and the others be the source domains, the proposed DCA is validated on dataset Office-31 by completing tasks $A, W \to D$; $A, D \to W$; $W, D \to A$; on dataset ImageCLEF-DA by completing tasks $I, C \to P$; $I, P \to C$; $C, P \to I$; on dataset Office-Caltech10 by completing tasks: $A, D, W \to C$; $C, D, W \to A$; $A, C, D \to W$, $A, C, W \to D$; and on dataset OfficeHome by completing tasks $A, C, P \to R$; $A, C, R \to P$; $A, P, R \to C$, $C, P, R \to A$.

4.4.2 Parameter Setting

This chapter employs $ResNet50$ as the shared feature extraction network ϕ, and all experiments are complemented by Pytorch. We add multi-scale kernels to the specific extraction layers used in previous studies (Zhu et al., 2019a,b; Li et al., 2021c) to collect multi-view features. The number of views in each source domain is 3. The parameters are updated based on back-propagation with Stochastic Gradient Descent (SGD), the momentum is 0.9, and the initial learning rate $\eta_0 = 0.01$. The learning rate of the shared network is one tenth of the other layers. Batch size $b = 32$, and the trade-off parameters $\alpha, \beta, \gamma, \lambda$ follow existing work (Zhu et al., 2019a), that is $\alpha = \beta = \gamma = \lambda = \frac{2}{1+exp(-10(e-1)/((I)))} - 1$, where e is training iteration, and $\mathcal{I} = 15000$ is the maximum iteration. Early stop is used to control the training progress. E for dataset Office-Caltech10 is 500, and for datasets ImageCLEF-DA, Office-31 and Office-Home, it is 2000. The threshold a_r in each category is the medium probability when selecting pseudo labels the first time, then the value is the maximum probability. The step q of adding pseudo labels in datasets ImageCLEF-DA, Office-31, Office=Caltech10 is 500, and in dataset Office-Home, it is 1000.

Generally, the learning rate η follows the same strategy in (Ganin and Lempitsky, 2015), which is $\eta = \frac{\eta_0}{(1+10(e-1)/((I)))^{0.75}}$. Since we add new samples to supervise the learning during training, for some tasks, an extra adjustment of the learning rate is needed when pseudo target labels are added for the first time. This adjustment aims to accelerate the convergence rate and reduce training time. From the Eth iteration, where we introduce pseudo labels for the first time, let the learning rate $\eta = \frac{\eta_0}{(1+10(e-E)/((I)))^{0.75}}$.

Taking datasets Office-31 and Office-Home as examples, the influence of adjusting the learning rate on the performance is shown in Tables 4.2, 4.3. “No” means without adjustment, “Yes” means with adjustment. It can be seen that adjusting the learning rate does not harm the transfer performance. For the target domains containing a few samples, the classification accuracy of the learning rate with adjustment is almost the same as that without adjustment. But when target domains have more samples, the adjustment can improve the accuracy. Hence, we define that when the number of target samples is larger than 1000, the extra learning rate is needed.

Figs. 4.3 shows the changes in the classification accuracy with the training

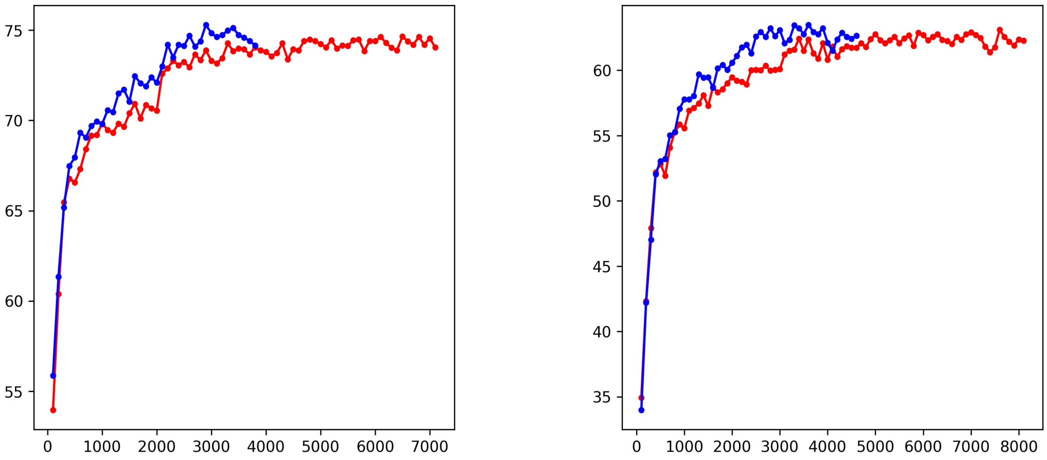
Table 4.2 : Accuracy (%) on dataset Office-31 with and without adjusting the learning rate when adding pseudo labels.

Standards	A, W→D	A, D→W	W, D→A	Avg
No	**99.6**	**98.9**	74.7	91.1
Yes	**99.6**	98.8	**75.1**	**91.2**

Table 4.3 : Accuracy (%) on dataset Office-Home with and without adjusting the learning rate when adding pseudo labels.

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
No	81.3	80.4	63.3	**72.1**	74.3
Yes	**81.4**	**80.5**	**63.6**	**72.1**	**74.4**

progress on datasets Office-31 and Office-Home with and without adjustment, taking tasks $W, D \rightarrow A$ and $A, P, R \rightarrow C$ as examples. The red line indicates the performance without adjustment, the blue line indicates the performance with adjustment. It shows that the accuracy with the learning rate adjustment increases more rapidly than that without adjustment. Tables 4.4 and 4.5 show the average training iterations needed for the tasks at the lowest test loss with and without adjusting the learning rate. The results indicate that the learning rate adjustment can reduce the training time. For dataset Office-31, the average training iteration reduces by nearly 30%, while for dataset Office-Home which has a large number of samples, the training iteration reduces by nearly 40% on average.



(a) Office-31: A

(b) Office-Home: C

Figure 4.3 : Accuracy (%) on target domains A from Office-31 and C from Office-Home as the training progresses. Red line means training without adjusting learning rate when adding pseudo labels, blue line indicates that with adjusting learning rate.

Table 4.4 : Training iterations on dataset Office-31 with and without adjusting the learning rate when adding pseudo labels.

Standards	A, W→D	A, D→W	W, D→A	Avg
No	2080	2260	5260	3200
Yes	**1780**	**2240**	**2820**	**2280**

Table 4.5 : Training iterations on dataset Office-Home with and without adjusting the learning rate when adding pseudo labels.

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
No	5080	6300	8540	5920	6460
Yes	**3160**	**6140**	**2640**	**3900**	**3960**

4.4.3 Results and Analysis

Tables 4.7, 4.6, 4.8 and 4.9 show the accuracy of the proposed and compared methods, where the highest accuracy is highlighted in bold. Validation standards include the performance of “Single best”, “Source Combine” and “Multi-Source”. “Single best” shows the performance of the source domain which achieves the highest accuracy using previous state-of-the-art domain adaptation methods with a single source; “Source Combine” shows the performance of some previous single source domain adaptation methods which mix multiple source domains into one; “Multi-Source” shows the performance of multi-source domain adaptation methods taking domain shifts into consideration.

The results indicate that the transfer performance of multiple source domains is generally superior to that of single source domains. Even when the data bias among source domains is disregarded, the models trained on mixed source domains achieve higher accuracy than most models trained on samples from a single domain. When taking source data bias into account and adapting each pair of source and target domains in specific feature spaces, the source models perform better on the target domain than adapting all domains in the same latent feature space.

Compared with previous studies employing multi-view features (MRAN and MSCLDA) and pseudo labels (ALDA), the accuracy of classifier alignment on most tasks is higher than the baselines. On several tasks, the compared methods achieve the best performance. The performance of MSCLDA on dataset ImageCLEF-DA is better than the proposed DCA. MSCLDA enriches transferable information at feature-level rather than classification-level. Feature-level concatenation generally requires extra constraint when extracting features, the proposed DCA adapts domain distributions under the supervision of pseudo-labels. It uses fewer loss functions when matching domains and extracting features, and outperforms the baselines on most tasks and datasets. Compared with other multi-source domain adaptation methods employing different combination rules, including average mean combination (MFSAN), weighted average mean combination based on perplexity scores (DCTN), weighted averaged mean combination with adjustment strategy (MSCLDA), adversarial learning based on combination (MADAN), the proposed method obtains the highest average accuracy on most datasets.

Table 4.6 : Accuracy (%) of the proposed and comparison methods on dataset Office-31.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Single best	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGard	99.1	96.9	68.2	88.1
	MRAN	99.8	96.9	70.9	89.2
	MDDA	99.2	97.1	73.2	89.8
	DDAN	**100.0**	96.7	65.3	87.3
	ALDA	**100.0**	97.7	72.5	90.1
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGard	99.7	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	MSCLDA	99.8	98.8	73.7	90.8
	DCA	99.6	**98.9**	**75.1**	**91.2**

Table 4.7 : Accuracy (%) of the proposed and comparison methods on dataset ImageCLEF-DA.

Standards	Method	I, C→P	I, P→C	P, C→I	Avg
Single best	ResNet	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGard	75.0	**96.2**	87.0	86.1
	MRAN	78.8	95.0	93.5	89.1
	MDDA	**79.8**	95.7	92.0	89.2
	DDAN	78.0	94.0	91.0	87.7
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGard	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN	79.1	95.4	93.6	89.4
	MSCLDA	79.5	95.9	**94.3**	**89.9**
	DCA	78.9	**96.2**	93.9	89.7

Table 4.8 : Accuracy (%) of the proposed and comparison methods on dataset Office-Caltech10.

Standards	Method	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
Single best	ResNet	82.5	91.2	98.9	99.2	93.0
	ADDA	88.8	94.5	99.1	98.0	95.1
	CyCADA	89.7	96.2	98.9	97.3	95.5
Source Combine	DAN	89.7	94.8	99.3	98.2	95.5
	ADDA	90.2	95.0	99.4	98.2	95.7
	CyCADA	91.0	95.9	99.0	97.8	95.9
Multi-Source	DCTN	90.2	92.7	99.4	99.0	95.3
	M3SDA	92.2	94.5	99.5	98.2	96.4
	MFSAN	93.8	95.1	99.1	98.7	96.7
	MSCLDA	94.1	95.3	99.1	98.5	96.8
	DCA	**94.7**	**96.0**	**99.7**	**99.1**	**97.4**

Table 4.9 : Accuracy (%) of the proposed and comparison methods on dataset Office-Home.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Single best	ResNet	75.4	79.7	49.6	65.3	67.5
	DAN	75.9	80.3	56.5	68.2	70.2
	D-CORAL	76.3	80.3	53.6	67.0	69.3
	RevGard	75.8	80.4	55.9	67.9	70.0
	MRAN	77.5	**82.2**	60.0	70.4	72.5
	MDDA	77.8	81.8	57.6	67.9	71.3
	DDAN	72.7	78.9	56.6	65.1	68.3
	ALDA	77.1	82.1	56.3	70.2	71.4
Source Combine	DAN	82.5	79.0	59.4	68.5	72.4
	D-CORAL	82.7	79.5	58.6	68.1	72.2
	RevGard	82.7	79.5	59.1	68.4	72.4
Multi-Source	MFSAN	80.8	79.0	60.7	70.0	72.6
	MADAN	81.5	78.2	54.9	66.8	70.4
	MetaMDA	**83.4**	81.2	60.5	70.2	73.8
	MSCLDA	80.6	79.9	61.4	71.6	73.4
	DCA	81.4	80.5	**63.6**	**72.1**	**74.4**

4.4.4 Comparison of Single-view Classifier and Multi-view Aligned Classifier

This section analyses the performance of single-view classifiers and multi-view aligned classifiers. To align view classifiers, alignment parameters are defined in each source domain. This chapter introduces the auxiliary classifier and the importance learning function to control the alignment parameters. The auxiliary classifier gives initial alignment parameter values over the batch, and outputs a linear combination of the multi-view classifiers. Taking this linear combination as the input of the importance function, the importance function learns the contribution of each view classifier and returns new alignment parameters which satisfy the alignment conditions mentioned in section 4.3.2 over the sample.

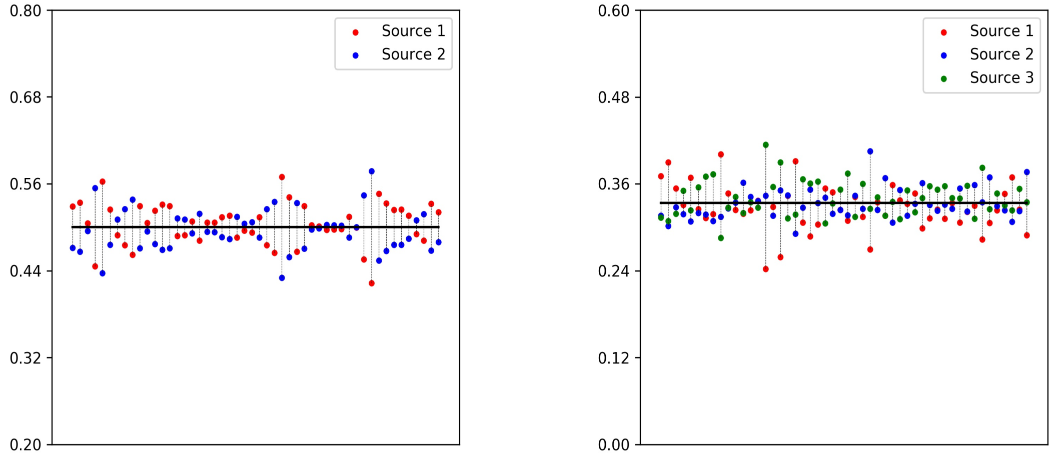
To validate the influence of the proposed alignment strategy, Table 4.10 shows the results of the experiments when calculating the alignment parameters using different strategies. Except for the proposed strategy, attention module SElayer (Hu et al., 2018b) and L_2 regularization are employed to determine the alignment parameters of multi-view classifiers. "None" indicates alignment without the importance learning function, "Attention" indicates the alignment parameters calculated using SElayer, "Regularization" indicates the alignment parameters optimized by L_2 regularization. Taking datasets Office-31 as an example, it can be seen that the proposed alignment parameter learning strategy achieves the highest accuracy. The proposed strategy can guarantee that the alignment parameters satisfy the mentioned conditions, and it has the superiority of fusing information from different aspects over other strategies.

Except for alignment parameters to align view classifiers, source combination parameters are used to combine source predictions when completing the target task. Combination parameters are calculated using the domain discriminator, the input is

Table 4.10 : Accuracy (%) of different alignment parameter calculation strategies on dataset Office-31

Standards	A, W-D	A, D-W	W, D-A	Avg
None	99.4	98.8	74.5	90.9
Attention	99.5	98.8	74.0	90.8
Regularization	99.4	**99.1**	74.1	90.9
Proposed	**99.6**	98.9	**75.1**	**91.2**

the sample feature extracted by the backbone, and the outputs are the probability of the target sample belonging to the source domains. To better understand the learning of source combination parameters, Fig. 4.4 displays source combination parameters by randomly selecting 50 samples from target domains A in Office-31 and A in Office-Home, respectively.



(a) Office-31: A

(b) Office-Home: A

Figure 4.4 : Combination parameters of target domains A from Office-31 and A from Office-Home.

Based on the learned alignment parameters and combination parameters, Tables 4.11, 4.12, 4.13 and 4.14 show the target classification performance of the source classifier trained on single view features and that of the aligned multiple classifiers. “C” indicates a single view classifier from a single source domain, “CA” indicates an aligned classifier in each source domain, “S” indicates the source domain (the source order is the same as described in Section 4.4.1), “M” indicates the combined multiple source classifiers, where “Mean” indicates the average combination, and “Weighted” indicates the proposed combination using the weights returned by the domain discriminator.

Commonly, when performing on the target domain, the single source aligned classifier outperforms the single view classifier, and the accuracy of the combined multi-source classifier is superior to the single source aligned classifier, and the weighted combination rule is superior to the average combination rule. The single view classifier performs similarly on the target domain in most situations, which may be a consequence of that the specific multi-view features we collected are based on the same pre-trained networks. For fair comparison, we employ commonly used network structures in previous studies to extract features and may take the robustness and heterogeneity of features as future study. The results of the aligned source classifier are higher than the single view classifier, which validates that employing the contributions of multi-view features has positive effects. When introducing multiple source classifiers, a noticeable growth in the accuracy can be seen on almost all datasets, and defining the importance of source domains can improve the performance.

4.4.5 Comparison of Feature Alignment and Classifier Alignment

Enriching feature-level information generally requires extra constraints such as orthogonality to eliminate redundancy, which may make the feature extraction more complex. Replacing feature alignment with classifier alignment can avoid the prob-

Table 4.11 : Accuracy (%) of classifier alignment and classifier trained on single view representations on dataset Office-31.

Standards		A, W→D	A, D→W	W, D→A	Avg
C1	S1	96.3	97.5	74.9	90.4
	S2	99.9	98.8	74.8	
C2	S1	96.3	97.8	74.8	90.4
	S2	**100.0**	98.8	74.9	
C3	S1	96.9	97.7	74.7	90.4
	S2	99.9	98.7	74.7	
CA	S1	97.1	97.8	74.8	90.6
	S2	99.9	**98.9**	75.0	
M	Mean	99.4	**98.9**	**75.1**	91.1
	Weighted	99.6	**98.9**	**75.1**	**91.2**

Table 4.12 : Accuracy (%) of classifier alignment and classifier with single view representations on dataset ImageCLEF-DA.

Standards		I, C→P	I, P→C	P, C→I	Avg
C1	S1	77.7	95.9	93.3	88.6
	S2	76.3	95.3	93.0	
C2	S1	77.4	95.9	93.6	88.8
	S2	76.7	95.5	93.5	
C3	S1	77.4	96.0	93.5	88.8
	S2	76.7	95.6	93.8	
CA	S1	78.1	96.0	93.7	89.0
	S2	77.0	95.7	93.6	
M	Mean	**79.0**	**96.2**	**93.9**	**89.7**
	Weighted	78.9	**96.2**	**93.9**	**89.7**

Table 4.13 : Accuracy (%) of classifier alignment and classifier with single view representations on dataset Office-Caltech10.

Standards		A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
C1	S1	94.3	95.4	98.8	98.0	96.7
	S2	94.0	95.0	98.5	97.2	
	S3	94.4	95.4	99.7	99.5	
C2	S1	94.0	95.4	98.7	97.6	96.7
	S2	94.2	95.2	98.4	97.3	
	S3	94.5	95.2	99.6	**99.9**	
C3	S1	94.1	95.5	98.6	97.8	96.7
	S2	94.4	95.3	98.4	96.9	
	S3	94.5	95.3	99.5	99.7	
CA	S1	94.2	95.4	98.8	98.0	96.7
	S2	94.4	95.2	98.3	97.1	
	S3	94.6	95.3	**99.7**	99.7	
M	Mean	**94.7**	95.9	**99.7**	99.2	**97.4**
	Weighted	**94.7**	**96.0**	**99.7**	99.1	**97.4**

Table 4.14 : Accuracy (%) of classifier alignment and classifier with single view representations on dataset Office-Home.

Standards		A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
C1	S1	80.4	77.7	62.4	70.3	72.6
	S2	79.5	78.5	62.2	70.0	
	S3	77.4	79.9	61.3	71.4	
C2	S1	80.2	77.7	62.4	70.2	73.0
	S2	79.3	78.2	62.2	70.3	
	S3	80.0	80.4	62.6	**72.4**	
C3	S1	80.5	77.7	62.6	70.3	73.0
	S2	79.5	78.1	62.4	70.0	
	S3	80.1	80.1	63.2	72.0	
CA	S1	80.5	77.9	62.7	70.3	73.2
	S2	79.6	78.1	62.4	70.2	
	S3	80.2	**80.5**	63.2	72.3	
M	Mean	81.3	79.3	**63.7**	72.0	74.1
	Weighted	**81.4**	**80.5**	63.6	72.1	**74.4**

lem caused by feature concatenation. Rather than considering what kind of information the multi-view features provide, we only need to ensure the predictions of multi-view classifiers are similar, which is advantageous when fusing the label information to obtain correct predictions with a high probability. To show the influence of feature alignment and classifier alignment, referring to feature alignment used in previous studies (Zhu et al., 2019b; Li et al., 2021c), Tables 4.15, 4.16, 4.17 and 4.18 compare the performance of a classifier trained on the aligned features with that of the aligned classifier. "FA" indicates training the classifier using concatenated multi-view features, "CA" indicates classifier alignment, "S" indicates a single source classifier, the source order is the same as described, and "M" indicates a multi-source classifier driven from a weighted combination rule.

It can be seen from the tables that classifier alignment results in higher performance than feature alignment, and the performance of the combined multi-source aligned classifier is usually better than the single source aligned classifier. It indicates that the classifier alignment which considers the importance of feature views significantly increases the performance compared with feature alignment which treats features from different views equally.

4.4.6 Influence of Pseudo Labels

This section explores the influence of pseudo labels, including when to add pseudo labels and the step involved in adding pseudo labels. Taking datasets Office-31 and Office-Home as examples, Tables 4.19 and 4.20 show the performance of the proposed method when adding pseudo labels at different iterations (the value of E), and Tables 4.21 and 4.22 show the performance of the proposed method when adding pseudo labels by different steps (value of q).

It indicates that with the training progresses, the probability of the pseudo labels being correct is becoming increasingly higher, resulting in an improvement in

Table 4.15 : Accuracy (%) of classifier alignment and feature concatenation on dataset Office-31.

Standards		A, W→D	A, D→W	W, D→A	Avg
FA	S1	97.8	97.9	73.4	90.1
	S2	99.6	98.5	73.5	
	M	**99.6**	98.5	73.5	90.5
CA	S1	97.1	97.8	74.8	90.6
	S2	99.9	**98.9**	75.0	
	M	**99.6**	**98.9**	**75.1**	**91.2**

Table 4.16 : Accuracy (%) of classifier alignment and feature concatenation on dataset ImageCLEF-DA.

Standards		I, C→P	I, P→C	P, C→I	Avg
FA	S1	**79.0**	95.5	93.4	89.2
	S2	78.7	95.1	93.6	
	M	78.9	95.6	93.7	89.4
CA	S1	78.1	96.0	93.7	89.0
	S2	77.0	95.7	93.6	
	M	78.9	**96.2**	**93.9**	**89.7**

Table 4.17 : Accuracy (%) of classifier alignment and feature concatenation on dataset Office-Caltech10.

Standards		A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
FA	S1	94.0	95.7	99.4	97.7	97.1
	S2	94.7	95.5	99.6	98.1	
	S3	94.7	95.6	**99.9**	**100.0**	
	M	**94.9**	95.9	99.8	99.5	**97.5**
CA	S1	94.2	95.4	98.8	98.0	96.7
	S2	94.4	95.2	98.3	97.1	
	S3	94.6	95.3	99.7	99.7	
	M	94.7	**96.0**	99.7	99.1	97.4

Table 4.18 : Accuracy (%) of classifier alignment and feature concatenation on dataset Office-Home.

Standards		A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
FA	S1	79.6	76.9	60.9	67.5	71.8
	S2	77.8	77.2	60.1	67.8	
	S3	80.1	**80.5**	61.9	70.7	
	M	**81.7**	79.6	62.5	71.3	73.8
CA	S1	80.5	77.9	62.7	70.3	73.2
	S2	79.6	78.1	62.4	70.2	
	S3	80.2	**80.5**	63.2	**72.3**	
	M	81.4	**80.5**	**63.6**	72.1	**74.4**

classification accuracy. When adding pseudo labels at the very start of training ($E = 100$), we are actually at a high risk of introducing the wrong labels to supervise the training, which leads to a degradation of the transfer ability on the target domain.

Table 4.19 : Accuracy (%) of adding pseudo labels at different epochs on dataset Office-31. Adding step is set as $q = 500$.

Standards	A, W→D	A, D→W	W, D→A	Avg
E=100	98.9	98.4	69.2	88.8
E=1000	99.4	98.8	74.1	90.8
E=2000	**99.6**	**98.9**	**75.1**	**91.2**

Table 4.20 : Accuracy (%) of adding pseudo labels at different epochs on dataset Office-Home. Adding step is set as $q = 1000$.

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
E=100	80.5	77.5	61.2	70.5	72.4
E=1000	81.2	78.9	63.3	71.3	73.7
E=2000	**81.4**	**80.5**	**63.6**	**72.1**	**74.4**

In a similar way, when adding pseudo labels using a small step ($q = 100$), the risk of introducing the wrong target labels is high, because target samples themselves can be divided into easy-to-predict samples and hard-to-predict samples. The sample belonging to the former cluster is given the same pseudo labels from the multiple source classifiers with high confidence, while the hard-to-predict sample in the latter cluster gets multiple pseudo labels or the same labels with low probability, and it is difficult for us to define which label is right. Hence, it is preferable to add pseudo labels using large steps. As shown in the tables, for dataset Office-31 with a few samples, $q = 500$ is appropriate. When the step becomes larger, the performance degrades, which means introducing pseudo labels when training becomes convergent may not make much difference. For dataset Office-Home with a large number of samples, $q = 2000$ has the highest average accuracy, meaning a large step is preferable for large dataset.

Table 4.21 : Accuracy (%) of adding pseudo labels using different steps on dataset Office-31. Start iteration is set as $E = 2000$.

Standards	A, W→D	A, D→W	W, D→A	Avg
q=100	99.3	98.6	73.6	90.5
q=500	**99.6**	**98.9**	**75.1**	**91.2**
q=1000	99.3	98.8	74.2	90.8

Table 4.22 : Accuracy (%) of adding pseudo labels using different steps on dataset Office-Home. Start iteration is set as $E = 2000$

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
q=100	80.3	77.7	62.1	70.8	72.7
q=500	**81.6**	80.1	63.9	71.7	74.3
q=1000	81.4	**80.5**	63.6	**72.1**	74.4
q=2000	**81.6**	80.3	**64.5**	**72.1**	**74.6**

4.4.7 Ablation Study

This section describes the ablation study of the proposed method on datasets Office-31 and Office-Home. We explore the influence of the main modules which directly affect the training and alignment of the classifiers, including domain adaptation (optimized by loss L_{adpt}), cross-domain constraint (reflected by loss L_{cro}), self-training module using pseudo target labels (parameterized by loss L_t), and classifier alignment (controlled by L_{ca}). Both the classification accuracy and MMD scores are employed as criteria to determine the effects of the modules. Tables 4.23-4.26 show the details. We use loss functions to denote corresponding network module, for example, standard "L_t" refers to the results returned by the model trained without self-training. "All" refers to all modules and constraints which are employed to train the model.

It can be seen from Tables 4.23 and 4.24 that domain adaptation and supervision of the pseudo target labels are the two most important constraints, since without either of these two modules, the classification accuracy decreases significantly. For dataset Office-31, the performance without self-training L_t is lower than that without domain adaptation L_{adpt}, meaning that the supervision of pseudo labels contributes more than domain adaptation. While for dataset Office-Home, the performance without self-training L_t is higher, indicating that domain adaptation L_{adpt} contributes more than pseudo labels. This is due to the number of pseudo labels with high confidence that we collected. For a dataset with only a few samples and categories, a small number of target samples can provide enough information to represent the target domain well. However, for a large dataset, a few pseudo-labeled target samples can enhance the learning but source samples are still necessary to dominate the classifier. The results without cross-domain constraint L_{cro} and classifier alignment L_{ca} are similar. Compared with results with all modules, these two modules also have a positive influence on ensuring the performance of the source classifiers on the target domain. Cross-domain constraint L_{cro} is helpful to collect pseudo labels with high probability, while classifier alignment L_{ca} is advantageous when fusing predictions from multiple aspects.

Table 4.23 : Accuracy (%) of different constraints on dataset Office-31.

Standards	A, W→D	A, D→W	W, D→A	Avg
L_t	99.4	90.0	72.1	87.2
L_{cro}	99.4	**98.9**	74.4	90.9
L_{adpt}	**99.8**	98.6	71.7	90.0
L_{ca}	99.4	98.8	74.5	90.9
All	99.6	**98.9**	**75.1**	**91.2**

Table 4.24 : Accuracy (%) of different constraints on dataset OfficeHome.

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
L_t	81.1	79.7	61.9	70.7	73.4
L_{cro}	**81.6**	79.6	**64.0**	71.1	74.1
L_{adpt}	81.2	78.8	61.7	69.1	72.7
L_{ca}	81.5	80.1	**64.0**	71.9	74.4
All	81.4	**80.5**	63.6	**72.1**	**74.4**

Tables 4.25 and 4.26 show the MMD scores of different modules. “C” indicates single view classifier, “S” indicates single source, the order is the same as described. High MMD scores mean large domain gaps. It can be seen that domain adaptation L_{adpt} is the most important module for reducing domain gaps, as without it, MMD scores gain the highest values, meaning the domain gap is the largest. MMD scores

without self-training L_t are affected by the noise of pseudo labels. Self-training contributes more than other modules when the noise is lower. Otherwise, classifier alignment L_{ca} which achieves higher MMD scores is more important. On some tasks, cross-domain constraint L_{cro} has a negative influence on reducing domain gaps as without this module the MMD scores achieve the lowest value. This is not surprising because the cross-domain constraint aims to reduce the gaps among multiple domains. If there is a large gap between a source and the target domains, other closer source domains can be affected when fitting the farthest one. But cross-domain constraint is helpful in collecting the correct pseudo labels, referring to the classification results, it has a positive effect on the classifiers.

4.4.8 Data Visualization

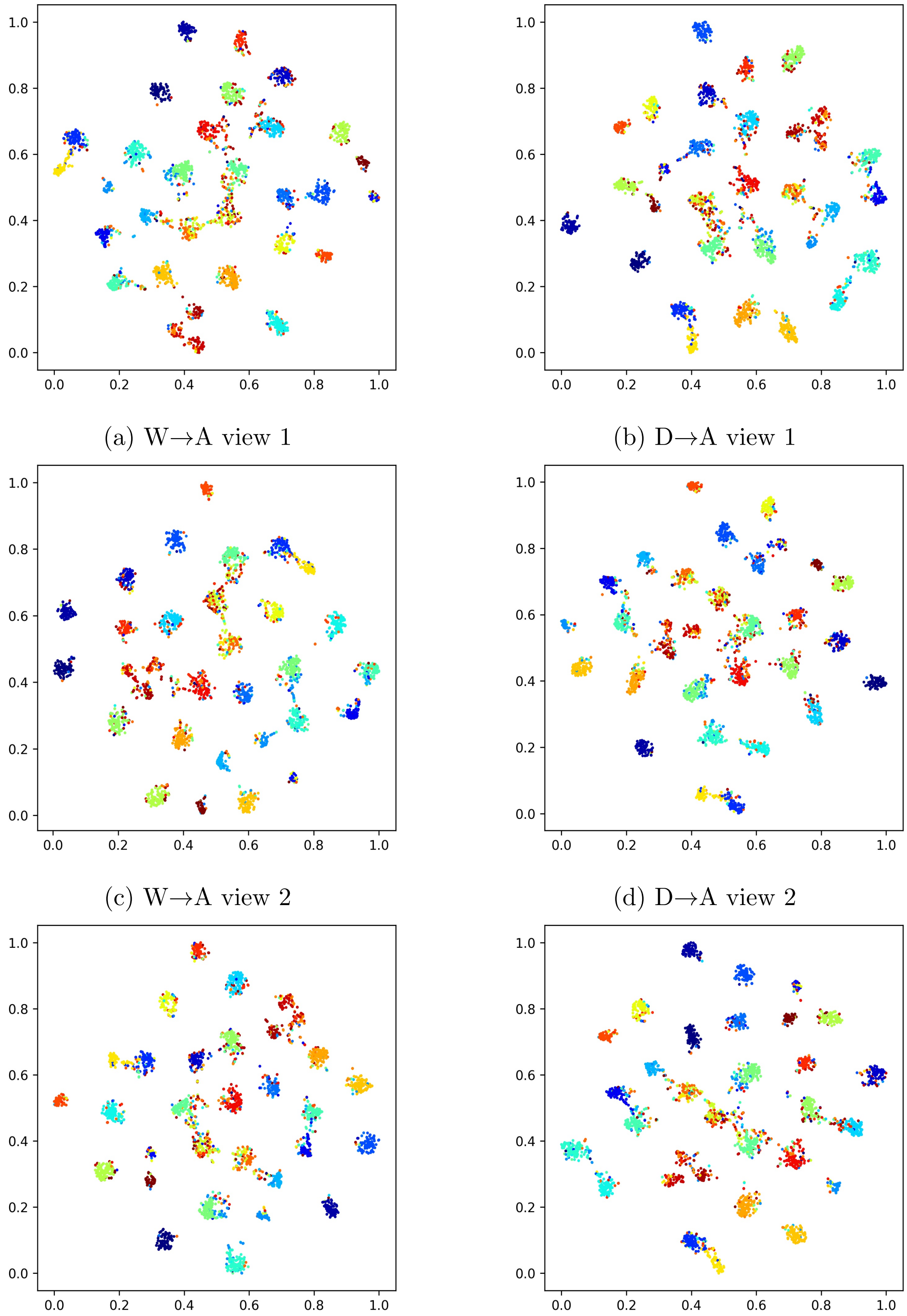
Taking task $W, D \rightarrow A$ from dataset Office-31 as an example, Fig. 4.5 shows the data visualization of the target samples in multiple feature spaces. Different colors indicate the categories. It can be seen that most samples are divided into correct classes with clear boundaries. Compared with views 1 and 2, features in the 3rd view have larger inter-class distance, and source D separates the target samples better than source W.

Table 4.25 : MMD scores (10^{-3}) of different constraints on dataset Office-31.

Standards		S1			S2		
		C1	C2	C3	C1	C2	C3
A, W→D	L_t	15.0	12.2	11.7	12.2	11.6	10.7
	L_{cro}	11.5	9.9	9.9	12.0	11.3	10.2
	L_{adpt}	13.0	11.3	11.6	12.1	11.6	10.8
	L_{ca}	**7.8**	**7.4**	**8.3**	10.7	10.4	9.9
	All	9.3	8.5	8.7	**10.2**	**9.6**	**9.7**
A, D→W	L_t	8.0	6.4	7.2	15.7	13.6	12.5
	L_{cro}	5.6	5.0	**5.8**	**12.1**	**11.5**	**11.0**
	L_{adpt}	8.4	7.3	7.8	15.6	13.7	12.7
	L_{ca}	6.1	5.3	5.9	13.4	12.1	11.3
	All	**5.3**	**4.9**	**5.8**	13.4	12.3	11.7
W, D→A	L_t	10.3	9.1	10.0	24.3	21.3	20.3
	L_{cro}	12.1	9.5	9.0	**13.4**	**12.9**	**12.9**
	L_{adpt}	60.6	45.6	37.2	64.3	53.4	44.0
	L_{ca}	9.1	8.0	9.3	18.1	16.7	16.2
	All	**7.5**	**6.9**	**7.2**	13.5	13.1	13.3

Table 4.26 : MMD scores (10^{-3}) of different constraints on dataset Office-Home.

Standards		S1			S2			S3		
		C1	C2	C3	C1	C2	C3	C1	C2	C3
A,C,P→R	L_t	5.8	6.1	5.2	9.8	8.9	6.7	16.6	15.0	10.8
	L_{cro}	6.7	5.6	5.3	4.3	4..3	**4.0**	7.8	7.2	5.9
	L_{adpt}	7.8	8.0	6.3	9.4	8.0	6.0	13.5	11.3	8.5
	L_{ca}	7.3	5.9	5.8	4.7	4.6	4.2	8.1	7.3	5.9
	All	**5.5**	**4.8**	**5.0**	**3.9**	**4.1**	**4.0**	**5.0**	**4.9**	**4.4**
A,C,R→P	L_t	**7.0**	**6.8**	**6.8**	7.7	7.5	6.6	8.4	7.5	6.3
	L_{cro}	11.2	9.5	9.2	**5.6**	**5.5**	**5.4**	3.8	3.8	3.7
	L_{adpt}	14.3	13.9	11.7	10.8	9.5	7.6	6.4	5.8	5.4
	L_{ca}	11.2	9.4	9.1	5.7	5.7	**5.4**	3.7	3.9	3.7
	All	13.1	10.4	9.4	6.9	5.7	6.2	**3.4**	**3.4**	**3.4**
A,P,R→C	L_t	**6.3**	**6.4**	**6.2**	13.7	12.7	10.2	11.9	12.9	9.1
	L_{cro}	10.5	8.5	8.4	**7.7**	**7.9**	**6.5**	5.1	5.5	**4.7**
	L_{adpt}	24.3	23.1	15.8	48.3	38.0	25.5	23.5	21.6	15.6
	L_{ca}	10.7	8.7	8.7	9.2	8.9	7.7	5.7	6.3	5.4
	All	9.1	7.8	8.1	8.1	8.3	7.2	**4.9**	**5.2**	5.0
C,P,R→A	L_t	10.5	9.5	8.5	21.6	18.4	15.3	23.4	19.2	14.5
	L_{cro}	**7.5**	**7.1**	**7.1**	**10.0**	**9.7**	**9.0**	9.3	7.7	6.9
	L_{adpt}	50.0	34.5	22.2	74.4	56.7	36.7	36.5	29.4	20.6
	L_{ca}	7.9	7.5	7.3	10.8	10.5	9.7	10.1	8.9	7.4
	All	8.5	8.4	7.9	11.8	11.3	10.3	**7.1**	**6.5**	**5.9**



(a) W→A view 1

(b) D→A view 1

(c) W→A view 2

(d) D→A view 2

(e) W→A view 3

(f) D→A view 3

Figure 4.5 : T-SNE visualization of multi-view features in target domain.

4.5 Summary

This section provides an overall summary of this work and offers suggestions for potential future study. This chapter proposes dynamic classifier alignment for multi-source domain adaptation, where the multi-view features and multi-source domains are investigated together to improve the transfer learning performance. To take advantage of the specific information carried by representations from different views, instead of concatenating multi-view features directly as in previous studies, we train multiple classifiers and align the multi-view classifiers to build the final source classifier with the assistance of a generated auxiliary source classifier. Compared with existing multi-source domain adaptation methods which combine the source domains averagely or calculate the combination weights relying on feature distance over batch, we develop a domain discriminator to learn the combination parameters with respect to the probabilities of a target sample belonging to the source domains. In addition, to explore more usable information from the target domain to enhance the cross-domain ability of the source classifier, pseudo labels are introduced during training to provide supervision. Experiments on four popular real-world image classification datasets show the proposed method achieves higher performance compared with most baselines.

The current study aligns classifiers from different views to avoid the feature heterogeneity problems of multiple views and domains. In real-world applications, not only the features spaces, but also the label space can be heterogeneous. Thus, based on this work, we will try to extend the alignment to a heterogeneous setting to make the algorithm more practicable in the future.

Chapter 5

Multi-Source Domain Adaptation with Sample and Source Distillation

5.1 Introduction

Multi-source domain adaptation attracts increasing attention as it delivers richer information. Most related works use all samples to train the transfer model but rarely consider the fact that some outliers or dissimilar samples might introduce additionally confusing information which degrades the performance. It is the quality of training samples, not just the quantity, that influences transfer performance. There might be source samples that relate weakly (inefficient samples) to the target samples which might confuse the classifier on both source and target tasks and result in negative transfer. As shown in Fig. 5.1, in relation to the class backpack, for example, although pictures with the dashed line from Source 1 and Source 2 contain information of backpack, the one in Source 1 provides extra unrelated information (people and bird) which may mislead the predictor, while the other in Source 2 can only provide incomplete information which may confuse the predictor. Samples from Source 3 are not real-world pictures which provide fewer details compared with other sources, meaning that Source 3 transfers inadequate knowledge to the target domain which contains real-world pictures.

To filter out unrelated information, distilling algorithms are proposed. Distant transfer algorithm builds a mixture of intermediate domains as a bridge to filter out unusable samples (Tan et al., 2017). Multi-source distilling network selects training samples using the estimated Wasserstein distance between a source sample and the

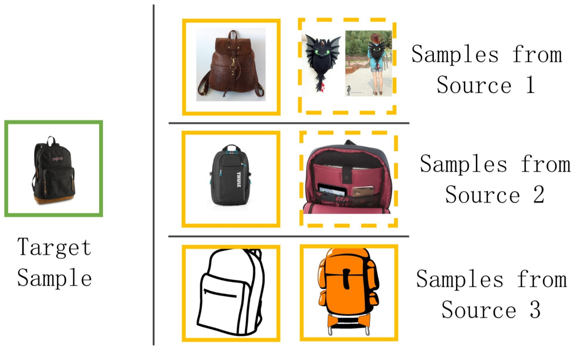


Figure 5.1 : Example of inefficient source samples and domains.

whole target domain (Zhao et al., 2020a). Partial feature selection and alignment network filters out unimportant feature dimensions and builds multiple adaptation losses to preserve both category-level and domain-level information (Fu et al., 2021).

However, the existing methods rely mainly on measuring the distance between the source samples and the whole target domain to select training samples close to the target domain, but disregard the correlation between source samples and the target classes, and the influence of boundary samples. Although these samples might be close to the target domain, they can cause misalignment. In addition, existing widely used combination rules based on feature distance fail to indicate source domains which are most related to the target domain.

To solve these problems, this chapter proposes a transfer sample and source distillation (SSD) method for multi-source domain adaptation, which develops a two-step selective strategy to distill the inefficient source samples as well as the domains based on the similarity between a source sample and the target category. Our contributions are threefold:

- We propose a two-step selective strategy to select transfer source samples and the dominant source domain. The proposed strategy can extract similar source knowledge that is more noticeable to the target domain. Simultaneously, it identifies the most similar source domain to guarantee the source predictor dominates the target prediction, which is of benefit in reducing negative transfer. This has rarely been considered in previous studies. The selection can also avoid misalignment resulting from outliers and preserves transferable information.

- We build an enhancement mechanism to improve the performance across domains of source predictors by adapting pseudo-labeled and unlabeled target samples, which explores the accessible target information directly except for domain feature matching on which most existing domain adaptation methods rely.
- We build a new combination rule to complete the target task. Compared with existing combination rules, the proposed rule not only estimates the combination weights but also identifies the dominant source domain to ensure the most similar source domain contributes the most in target task prediction, which previous multi-source domain adaptation methods disregard.

5.2 Problem setting and Notations

We focus on domain adaptation with multiple sources under a homogeneous setting, where the source and target domains have the same feature space and share the same label space. The proposed sample and source distillation method is evaluated on real-world image classification tasks.

Table 5.1 displays the notations and corresponding descriptions used in this chapter.

Table 5.1 : Notations and descriptions.

Notation	Description
$\mathcal{D}_{s_k}$, $\mathcal{D}_t$	source/target domain, k is source index
$\mathcal{D'}_{s_k}$	distilled source domain
$\mathcal{D}_{t_l}$	target domain with selected pseudo labels
$\mathcal{D}_{t_u}$	target domain with unlabeled samples
n_{s_k}, n_t	number of samples from source/target domain
$\boldsymbol{x}_{s_k}$, $\boldsymbol{x}_t$	sample from the source/target domain
$\boldsymbol{y}_{s_k}$	corresponding label of $\boldsymbol{x}_{s_k}$
ϕ	pre-trained backbone
ϕ_k	feature extractor for kth source domain
P_{s_k}	the kth source predictor
P_c	category classifier learned from pseudo-labeled target domain
P_d	domain discriminator learned from distilled source domains

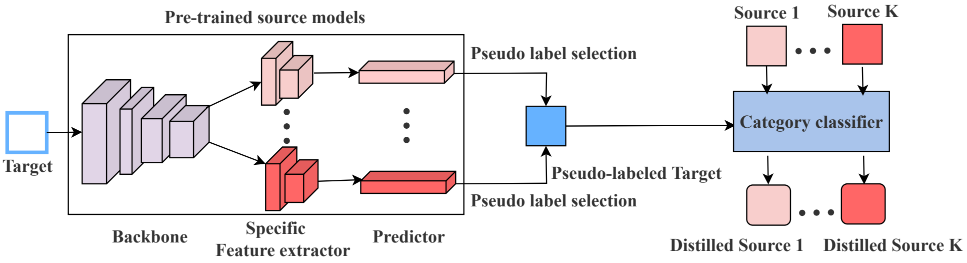
5.3 The Proposed Method: Sample and Source Distillation

The proposed method is illustrated in Fig. 5.2. As shown in Fig. 5.2(a), first, given pre-trained source models which are learned only based on source data, a portion of target labels being correct with a high probability are collected by ranking target predictions when applying target data to source predictors. Then, the category classifier is learned from the pseudo-labeled target domain, which is expected to select source samples highly related to the target domain and distill inefficient samples. Only source samples strongly connected to the target domain are kept for further training. Fig. 5.2(b) shows domain adaptation based on distilled source domains. First, a domain discriminator is learned from the selected source samples which defines the dominant source domain given its similarity to the target domain. At the same time, it returns the alignment weights for merging source predictions to complete the classification in target domain. Simultaneously, an enhancement mechanism based on self-supervised learning is built to improve the performance across domains of source models, where the selected target pseudo labels are adopted to parameterize the training. In addition, multi-level distribution matching is adopted to fine-tune the source models based on the selected source samples, which fits source models to the target domain by reducing data gaps. The target task is finally completed using the combination weights estimated by the learned domain discriminator. The operation of selecting highly similar source samples in Fig. 2(a) and the dominant source domain in Fig. 2(b) is named as a two-step selective strategy.

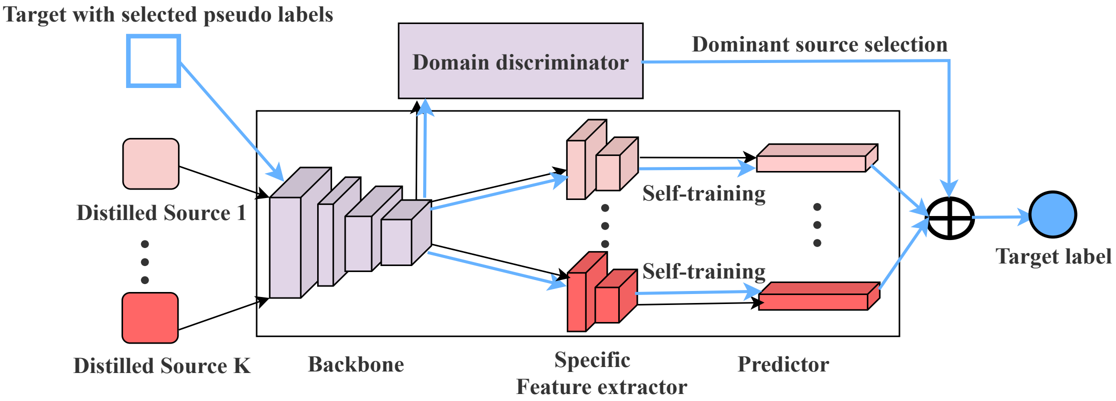
5.3.1 Source Model Training and Pseudo Target Label Collection

In terms of the structural risk minimization principle (Vapnik and Vapnik, 1998), the learning processing of the kth source predictor P_{s_k} can be expressed as:

$$P_{s_k} = \mathop{\arg\min}_{\substack{P_{s_k} \in \mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \sim \mathcal{D}_{s_k}}} L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{y}_{s_k}), \tag{5.1}$$



(a) Target pseudo label collection and source sample distillation.



(b) Domain adaptation and target task completion.

Figure 5.2 : The whole framework of the proposed method. Figure (a) indicates target label collection and source sample distillation. Given pre-trained source models comprised of shared backbone, specific feature extractors and predictors, pseudo target labels are collected first. The selected target labels are then employed to train the category classifier which distills inefficient source samples. Figure (b) indicates domain adaptation and dominant source domain selection. A domain discriminator is first built based on the distilled source domains to identify the dominant source domain and learn the relationships between source and target domains. Simultaneously, pre-trained source models are fine-tuned based on the distilled source domains by minimizing the discrepancy between the source and target distributions. Self-supervised training is also adopted based on the selected target labels to enhance the cross-domain ability of source models and guarantee the performance of transfer.

L is the cross-entropy loss function on labeled source data, which can be formulated as:

$$L = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}^i))), \tag{5.2}$$

ϕ, ϕ_k are feature extraction networks, $\mathcal{H}$ represents reproducing kernel Hilbert space and n_s can be replaced with the batch size as the training progresses over the batch. This replacement is applied to all update processing.

To improve the cross-domain ability of each source classifier, cross-domain constraint among source domains is adopted to ensure the generality of source classifiers by jointly training source models simultaneously. For the kth source domain, the jointly training with other source classifiers is formulated as:

$$\mathcal{L}_{cro} = \frac{1}{(K-1)} \sum_{k' \neq k}^{K} L(P_{s_{k'}}(\phi_{k'}(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{y}_{s_k}). \tag{5.3}$$

P_{s_k} in equation (5.1) can be re-written as:

$$P_{s_k} = \underset{\substack{P_{s_k} \in \mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \sim \mathcal{D}_{s_k}}}{\arg\min} \; L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{y}_{s_k}) + \beta L_{cro}. \tag{5.4}$$

Then pseudo target label is predicted as:

$$\hat{\boldsymbol{y}}_t = \wedge(P_{s_1}(\phi_1(\phi(\boldsymbol{x}_t))), \cdots, P_{s_K}(\phi_K(\phi(\boldsymbol{x}_t)))), \tag{5.5}$$

$\wedge$ is an operation to return the target labels predicted as the same by multiple source predictors. This operation aims to collect pseudo labels with low noise, which is important to guarantee the performance of source sample selection. If a target sample obtains the most votes from source predictors, it will be selected to train the distillation model. Otherwise, it will not be regarded as a transfer sample. In addition, a threshold a_t^c of each class is established to choose the pseudo labels with high probabilities. This can also help reduce the label noise. In this chapter, a_t^c is defined as the median value of the target samples which are predicted to belong to the same class. If no selected target samples belong to a class, we think there are very large gaps in this class among all source and target domains. In this situation, it is preferable to keep all source samples without distillation to gain enough class information.

5.3.2 The Two-step Selective Strategy

To select the source samples which are strongly connected to the target domain, the pseudo-labeled target domain is used to learn a series of category classifiers that discriminate transfer and inefficient source samples in each category. Category classifier, which is composed of multiple binary classifiers $\{P_c\}_{c=1}^{C}$, is trained by minimizing the cross-entropy loss:

$$L_{bce} = \sum_{c=1}^{C} L(P_c(\phi(\boldsymbol{x}_t)), I(\hat{\boldsymbol{y}}_t, c)). \tag{5.6}$$

$\hat{\boldsymbol{y}}_t$ is pseudo target label, $I(\hat{\boldsymbol{y}}_t, c) = 1_{\hat{\boldsymbol{y}}_t = c}$. Feed the kth source samples into the learned classifiers, the prediction of a source sample belonging to a target category is:

$$I(\hat{\boldsymbol{y}}_{s_k}, c) = P_c(\phi(x_{s_k})). \tag{5.7}$$

Compare predictions of sample x_{s_k} returned by c classifiers, if $I(\hat{\boldsymbol{y}}_{s_k}, c) = 1$ gains the highest probability value, denote $\hat{\boldsymbol{y}}_{s_k} = c$. If the ground-truth label $\boldsymbol{y}_{s_k} = c$, the source sample is regarded as a similar sample to the target domain. Otherwise, it will be removed from the corresponding source domain. After collecting similar source samples, to ensure that we select the source samples most related to the target domain, we rank the similar samples according to their probability values predicted by the corresponding binary classifier, and choose the top half of source samples as transfer samples to re-train the source model to fit the target domain.

Considering the label noise or scarcity in the pseudo-labeled target domain, to avoid that there is no source sample belonging to a target category, denote the probability vector of a source sample belonging to the category as $\boldsymbol{p}_{s_k} = [p_{s_k}^1, p_{s_k}^2, \cdots, p_{s_k}^C,]$, where $p^c_{s_k}$ is the maximum probability returned by the cth category classifier, a threshold a^c of each class is defined to guarantee at least one source sample in each category is selected as a transfer sample. The selective rule for a source sample $\boldsymbol{y}_{s_k} = c$ is: if $p^c_{s_k} > a_c$, then the source sample is kept for transferring. a_c is the median probability value of source samples from the cth class which are distilled by category classifier in the first selection. This ensures that source samples that gain a higher agreement of belonging to their category can be selected.

The source sample selection based on the assumption of the category classifier learned from pseudo-labeled target samples can extract more noticeable source information to the target domain, since the category classifier learns both invariant and specific information from the target domain.

Denote source domains with transfer samples as $\{\mathcal{D}'_{s_k}\}_{k=1}^{K}$, a domain discriminator P_d is trained to learn the degrees of a sample belonging to the source domains. It can be expressed as:

$$L_{dce} = \sum_{k=1}^{K} L(P_d(\phi(\boldsymbol{x}_{s_k})), k), \boldsymbol{x}_{s_k} \in \{\mathcal{D}'_{s_k}\}_{k=1}^{K}, \tag{5.8}$$

k is the domain label. Relatedness between target sample and source domains is defined as:

$$\boldsymbol{p}_{t_d} = P_d(\phi(\boldsymbol{x}_t)) = [p^1_{t_d}, p^2_{t_d}, \cdots, p^K_{t_d}]. \tag{5.9}$$

The maximum element of $\boldsymbol{p}_{t_d}$ indicates target sample belongs to the corresponding source domain.

By applying domain discriminator to a target sample, the similarity between a target sample and the whole source domain can be found. Then the predictor from this nearer source domain will gain a larger weight when predicting the target label by combining all source predictions. If there is a large portion of target samples showing high similarity to the same source domain, this source domain will be regarded as dominant source domain, whose prediction will be taken as the final prediction of target domain. The selection of the dominant source domain has the advantage of reducing negative transfer resulting from some unrelated source domains by giving the dominant source predictor a very large combination weight.

Let the number of the target samples belonging to the kth source domain be n_k, define if $\frac{n_k}{n_t} - \frac{n_t - n_k}{n_t} > \frac{1}{K}$, the kth source domain is a dominant domain for the target task.

5.3.3 Domain Adaptation and Target Task Completion

When completing the target task, pre-trained source models are re-trained using the selective transfer source samples by adapting source and target domains. Even though we have the most similar source samples/domains, there are still data gaps. To ensure that the source predictors can perform reasonably well, source and target data are adapted on both domain-level and class-level. It can be expressed as:

$$L_d(\mathcal{D}'_{s_k}, \mathcal{D}_t) = \underset{\psi \in \mathcal{H}}{\mathcal{MMD}}(\mathcal{D}'_{s_k}, \mathcal{D}_t) \tag{5.10}$$

and

$$\begin{split} L_c(\mathcal{D}'_{s_k}, \mathcal{D}_t) = \frac{1}{C} \sum_{c=1}^{C} \underset{\psi \in \mathcal{H}}{\mathcal{MMD}}(\mathcal{D}'^{c}_{s_k}, \mathcal{D}^c_t) - \\ \frac{1}{2C(C-1)} \sum_{c_1=1}^{C} \sum_{c_2 \neq c_1}^{C} (\underset{\psi \in \mathcal{H}}{\mathcal{MMD}}(\mathcal{D}'^{c_1}_{s_k}, \mathcal{D}'^{c_2}_{s_k}) + \\ \underset{\psi \in \mathcal{H}}{\mathcal{MMD}}(\mathcal{D}^{c_1}_t, \mathcal{D}^{c_2}_t)), \end{split} \tag{5.11}$$

where L_d is the divergence of adapting distributions on domain-level, L_c indicates that on class-level. $\mathcal{D}'^{c}_{s_k}, \mathcal{D}^c_t$ denote domains only containing the cth class samples. MMD (Pan et al., 2010) is formulated as:

$$\begin{split} &\underset{\psi \in \mathcal{H}}{\mathcal{MMD}}(\mathcal{D}_1, \mathcal{D}_2) \\ &= \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \psi(\phi_k(\phi(\boldsymbol{x}_1^i))) - \frac{1}{n_2} \sum_{j=1}^{n_2} \psi(\phi_k(\phi(\boldsymbol{x}_2^j)) \right\|^2_{\mathcal{H}}, \end{split} \tag{5.12}$$

ψ is a nonlinear function transforming data into RKHS with a universal kernel $\mathcal{K}$ satisfying $\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \psi(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2) \rangle$, n_1, n_2 are numbers of samples which can be replaced with batch size during training.

At the same time, cross-domain constraint is applied on target domain to reduce the misalignment of boundary samples by minimizing the error of results returned by all source predictors on the same target samples. The loss function in equation (5.3) is re-written as:

$$\begin{aligned}\mathcal{L}_{cro} = \frac{1}{(K-1)} \sum_{k' \neq k}^{K} \\ (\frac{1}{n_t} \sum_{j=1}^{n_t} \left| P_{s_k}\big(\phi_k(\phi(\boldsymbol{x}_t^j))\big) - P_{s_{k'}}\big(\phi_{k'}(\phi(\boldsymbol{x}_t^j))\big) \right|).\end{aligned} \tag{5.13}$$

Motivated by previous study (Li and Hospedales, 2020) which splits source domain, an enhancement strategy is built to improve the cross-domain ability of source predictors.

Dividing the pseudo-labeled target domain into $\mathcal{D}_{t_l}$ and $\mathcal{D}_{t_u}$, $\mathcal{D}_{t_u}$ contains the target samples gaining low probabilities when they are pseudo labeled by the pre-trained source predictors, of which the probabilities are lower than a_c^t, while $\mathcal{D}_{t_l}$ contains samples reaching probabilities higher than a_c^t. For the kth source domain, the constraint based on the pseudo labeled target domain is expressed as:

$$L_{supt} = \sum_{k=1}^{K} L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{t_l}))), \hat{\boldsymbol{y}}_{t_l}) + \delta L_d(\mathcal{D}_{t_l}, \mathcal{D}_{t_u}). \tag{5.14}$$

L_d is the adaptation loss, L is the cross-entropy loss of pseudo-labeled target domain, δ is the trade-off parameter.

The total loss is:

$$\begin{aligned}P_{s_k} = \underset{\substack{P_{s_k} \in \mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \sim \mathcal{D}'{}_{s_k}}}{\arg\min} \quad L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{y}_{s_k}) \\ + \alpha(L_d + L_c) + \beta L_{cro} + L_{supt}.\end{aligned} \tag{5.15}$$

When completing target task, if the dominant source domain exists, the target predictor will be the dominant source predictor (denote as $P_{s_{dmnt}}$). Otherwise, the target predictor is a weighted combination of the source predictors, and the weights are returned by the domain discriminator in equation (5.9). The target predictor is formulated as:

$$P_t = \begin{cases} \sum_{k=1}^{K} p_{t_d}^{k} \cdot P_{s_k}, \text{ if dominant domain is false;} \\ P_{s_{dmnt}}, \text{ if dominant domain is true.} \end{cases} \tag{5.16}$$

The processing of the transfer sample selection is described in Algorithm 4, and the target task prediction is outlined in Algorithm 5.

Algorithm 4 Transfer sample selection

```
**Input:** Source domain $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$, target domain $\mathcal{D}_t$;
**Initialization:** Feature extraction networks $\phi$, $\phi_k$ and source predictor $P_{s_k}$;
**for** $\epsilon = 1, \epsilon < \mathcal{I}_1, \epsilon + +,$ **do**
    Enhance cross-domain ability as in (5.3);
    Train source model as in (5.4);
**end for**
Calculate pseudo target label $\hat{\boldsymbol{y}}_t$ as in (5.5);
Learn category classifiers as in (5.6);
Select transfer source samples as in (5.7);
**Output:** Distilled source domains $\{\mathcal{D}'_{s_k}\}_{k=1}^{K}$.
```

Algorithm 5 Target task prediction

```
Input: Distilled source domains $\{\mathcal{D}'_{s_k}\}_{k=1}^{K}$, target domain $\mathcal{D}_t$, pseudo labeled training domain $\mathcal{D}_{t_l}$, and unlabeled domain $\mathcal{D}_{t_u}$. Pre-trained feature extraction networks $\phi$, $\phi_k$ and source predictor $P_{s_k}$;
Learn domain discriminator as in (5.8);
Calculate the weight vector $\boldsymbol{p}_{t_d}$ of the target sample belonging to source domains as in (5.9);
Select source dominant domain if exists;
for $\epsilon = 1, \epsilon < \mathcal{I}_2, \epsilon + +,$ do
    Adapt source and target data by minimizing MMD as in (5.10) and (5.11);
    Enhance cross-domain ability as in (5.13);
    Fit source model to target domain using self-supervised enhancement strategy as in (5.14);
    Update $\phi$, $\phi_k$ and $P_{s_k}$ as in (5.15);
end for
Complete target predictor as in (5.16);
Output: Target labels.
```

5.4 Experiments

In this section, the proposed transfer sample selection method is validated on five popular real-world visual datasets, comprising ImageCLEF-DA, Office31, Office-Caltech10 and OfficeHome. All the experiments are classification tasks under the multi-source domain adaptation scenario. Classification accuracy is the only criterion used to evaluate the performance. The dataset details, parameter settings, experiment results and the analysis are detailed in the following.

5.4.1 Datasets and Baselines

Five real-world datasets are used in this chapter to valid the proposed SSD. ImageCLEF-DA has three domains sharing 12 categories, the proposed method is applied by completing three tasks: $I, C \to P$; $I, P \to C$; $C, P \to I$. Office-Caltech10 contains four domains sharing 10 categories. The proposed method is validated by completing four tasks: $A, D, W \to C$; $C, D, W \to A$; $A, C, D \to W$, $A, C, W \to D$. Office-31 comprises three domains and contains 4110 images which share 31 categories. The proposed method is tested by completing three tasks: $A, W \to D$; $A, D \to W$; $W, D \to A$. Office-Home holds 15588 images which share 65 categories. Experiments are conducted by completing four tasks: $A, C, P \to R$; $A, C, R \to P$; $A, P, R \to C$; $C, P, R \to A$. DomainNet is the largest dataset containing 0.6 million images sharing 345 categories. The proposed method is tested on six tasks. For each task, two unrelated source domains defined by the domain discriminator are distilled, the target tasks are finally predicted using three selected domains: $I, R, S \to C$, $P, Q, R \to I$, $C, I, R \to P$, $C, I, S \to Q$, $I, Q, S \to R$ and $C, P, Q \to S$. All results of the compared multi-source domain adaptation methods are predicted using five source domains, the proposed method uses three selected source domains.

Baselines with single source domain include:

- DAN: Deep adaptation network employing joint distribution merging (Long et al., 2015);
- RevGrad: Reverse gradient based adaptation (Ganin and Lempitsky, 2015);
- D-CORAL: Adaptation employing correlation alignment (Sun and Saenko, 2016);
- CyCADA: Adaptation based on cycle-consistent adversarial learning (Hoffman et al., 2018);
- MRAN: Adaptation with multi-view representations (Zhu et al., 2019b);
- CAT: Teacher guided adaptation with cluster alignment (Deng et al., 2019);
- SAFN: Adaptive feature norm adaptation (Xu et al., 2019a);
- MDE: Deep network with minimum discrepancy estimation (Rahman et al., 2020);
- ETD: Adaptation with enhanced transport distance (Li et al., 2020c);
- FDA: Faster domain adaptation network (Li et al., 2021b);
- DWL: Dynamic weighted learning for unsupervised domain adaptation (Xiao and Zhang, 2021);
- CRSL: Adaptation employing cycle-reconstructive subspace learning (Xu and Yan, 2022).

Multi-source domain baselines include:

- MFSAN: Deep network combining feature and classifier alignment (Zhu et al., 2019a);
- FADA: Adaptation with federated adversarial learning (Peng et al., 2019b);

- MDDA: Adaptation with source distillation (Zhao et al., 2020a);
- MADAN: Multi-source adaptation with adversarial aggregation network (Zhao et al., 2021);
- MetaMDA: Meta-learning based adaptation employing an online optimization strategy (Li and Hospedales, 2020);
- MIAN: Multi-source domain adaptation with information-theoretic regularization (Park and Lee, 2021);
- WBT: Multi-source domain adaptation with Wasserstein barycenter transport (Montesuma and Mboula, 2021);
- MSCLDA: Multi-level and multi-view adaptation with source contribution learning (Li et al., 2021c);
- MLAN: Joint and separate adaptation with mutual learning (Xu et al., 2022);
- DCA: Multi-view adaptation with sample-wise classifier alignment (Li et al., 2022a).

5.4.2 Parameter Setting

Our experiments employ $ResNet50$ as the backbone on datasets Office-31, ImageCLEF-DA and Office-Home, $ResNet101$ as the backbone on datasets Office-Caltech10 and DomainNet, complemented by Pytorch. Parameters are updated based on back-propagation with Stochastic Gradient Descent (SGD), the momentum is 0.9, the learning rate η follows the same strategy in (Ganin and Lempitsky, 2015), which is $\eta = \frac{\eta_0}{(1+10\epsilon)^{0.75}}$, where $\eta_0 = 0.01$, ϵ is the training progress changing linearly from 0 to 1. The learning rate of the shared network is one tenth of other layers. Batch size $b = 32$, trade-off parameters α, β, δ follow existing work (Zhu et al., 2019a), that is $\alpha = \beta = \delta = \frac{2}{1+exp(-10\epsilon')} - 1$, where ϵ' is a linearly changing number from 0 to 1. Early stop is used to control the training process. Considering the large memory requirement, for class-level distribution matching, we only enlarge the distance between the nearest classes, not them all.

Threshold a^c is defined to guarantee that each source category contains at least one training sample. We choose the medium value of the predicted probabilities in each category as the threshold in selecting the transfer source samples. Threshold a_t^c is the median value of traget samples belonging to the same class predicted by source classifiers.

5.4.3 Results and Analysis

Tables 5.2, 5.3, 5.4, 5.5 and 5.6 show the performance of what we propose (SSD) and the baselines under three standards: "Single best", "Source Combine" and "Multi-Source". "Single best" shows the best performance of some state-of-the-art single source domain adaptation methods; "Source Combine" displays the performance of several compared single source domain adaptation methods using multiple source domains where all source samples are mixed; "Multi-Source" lists the performance of multi-source domain adaptation methods which consider domain shifts. Comparison with the single source domain adaptation method considers two aspects: first, it aims to show that the proposed method can avoid negative transfer resulting from combining multiple source predictions. Second, if the dominant source domain exists, the comparison reveals the superiority of the proposed method based on single source domain. The text in bold indicates the highest accuracy of each task.

It shows that the predictors learned with the transfer samples achieve the highest average performance on most datasets. The performance of multi-source domain adaptation under both standards -"Source Combine" and "Multi-Source" -is usually better than that of single source domain adaptation, and the performance of

“Multi-Source” is generally superior to what is achieved with the “Source Combine”. This means richer training samples can enhance the learning processing in most situations. When considering the domain shifts between source domains, the learned predictor can perform better than what can be obtained by simply combining all training samples. The proposed method trained with selective samples and domains outperforms most existing methods trained with all source samples. It indicates that the quantity of the training samples is not the only factor that affects the performance. The sample quality is more important. Distilling the inefficient training samples even domains does not degrade the learning performance in most situations. On dataset ImageCLEF-DA, baseline DWL gains significant superiority over other methods. DWL dynamically weights all source samples to ensure similar source samples control domain alignment. Compared with DWL, the proposed method uses fewer source samples, and outperforms the baseline on other datasets.

The compared multi-source domain adaptation methods involve the widely used combination rules, including the averaged combination (MFSAN); the weighted mean combination (DCTN, MDAN), in which the weight is calculated using the distance between the source and target domains; and the adversarial learning strategy (MADAN). Results indicate that the proposed method with the developed combination rule, which takes the dominant source domain into account, achieves higher performances on most target tasks than the other methods compared.

Compared with multi-source domain adaptation method which considers both the importance of source domains and source samples (MDDA), the proposed mehtod (SSD) achieves higher performance on most tasks and the highest average performance. MDDA distills source samples based on the Wasserstein distance between a source sample and the whole target domain, and calculates source weights using standard Gaussian distribution to combine source predictions. The proposed SSD distills source samples by measuring the distance between a source sample and the target category, and automatically learns the source importance by designing a domain discriminator to accept a target sample as a source insider. It can align source information that is more noticeable to the target domain.

5.4.4 Influence of the Combination Rule

The proposed combination rule is related to the quantities of the target samples belonging to the source domains, and for each target domain, there might be a dominant source domain. Fig. 5.3 shows how many target samples are divided into each source domain. The source rank is consistent with the description: for instance, for task $A, D \rightarrow W$, "Source 1" (S1) indicates domain A, "Source 2" (S2) indicates domain D. If the quantity of the target samples belonging to a source domain exceeds the threshold, which is $n_k > \frac{(K+1) \cdot n_t}{2K}$, as mentioned in 5.3.2, the corresponding kth source domain is a dominant source domain. Calculations illustrate that, for target tasks $I, C \rightarrow P$ from ImageCLEF-DA, $A, W \rightarrow D$, $A, D \rightarrow W$ from Office31, $A, C, W \rightarrow D$ from Office-Caltech10 and $C, P, R \rightarrow A$, $A, C, R \rightarrow P$ from OfficeHome, the dominant source domains exist. For DomainNet, although dominant source domain does not exist, the source domains gaining very low weights can be distilled. For example, when predicting target task Q, Sources 3 and 4 are distilled as there are fewer target samples belonging to these two source domains, meaning they are weakly-connected to the target domain.

Tables 5.7, 5.8, 5.9, 5.10 and 5.11 show the performance of the proposed method with different combination rules. "S" is the performance of single source domain adaptation with cross-domain constraint. "Mean" indicates the performance of multi-source domain adaptation using an averaged combination. "Weighted" indicates the performance of multi-source domain adaptation using a weighted mean combination, the source weights are calculated as shown in equation (5.9). "Proposed" indicates the performance of multi-source domain adaptation using the pro-

Table 5.2 : Accuracy (%) on dataset Office31 of the proposed and comparison methods

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
Single	RevGard	99.1	96.9	68.2	88.1
best	MRAN	99.8	96.9	70.9	89.2
	CAT	**100.0**	986	70.4	89.7
	SAFN	99.8	98.4	69.8	89.3
	ETD	**100.0**	**100.0**	71.0	90.3
	FDA	**100.0**	99.1	74.3	91.1
	MLAN	99.6	98.9	75.7	91.4
	DWL	**100.0**	99.2	73.1	90.8
Source	DAN	99.6	97.8	67.6	88.3
Combine	D-CORAL	99.3	98.0	67.1	88.1
	RevGard	99.7	98.1	67.6	88.5
	MFSAN	99.5	98.5	72.7	90.2
Multi-	MIAN	98.5	99.5	74.7	90.9
Source	MSCLDA	99.8	98.8	73.7	90.8
	DCA	99.6	98.9	75.1	91.2
	SSD	99.8	99.1	**76.0**	**91.6**

Table 5.3 : Accuracy (%) on dataset ImageCLEF-DA of the proposed and comparison methods

Standards	Method	I, C→P	I, P→C	P, C→I	Avg
Single best	ResNet	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGard	75.0	96.2	87.0	86.1
	MRAN	78.8	95.0	93.5	89.1
	CAT	76.7	97.9	93.3	89.3
	SAFN	78.0	96.2	91.7	88.6
	ETD	81.0	97.9	93.3	90.7
	FDA	79.2	97.2	93.0	89.8
	DWL	**82.3**	**98.1**	**94.8**	**91.7**
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGard	77.9	93.7	91.8	87.8
Multi-Source	MFSAN	79.1	95.4	93.6	89.4
	MSCLDA	79.5	95.9	94.3	89.9
	DCA	78.9	96.2	93.9	89.7
	SSD	79.2	96.6	**94.8**	90.2

Table 5.4 : Accuracy (%) on dataset Office-Caltech10 of the proposed and comparison methods

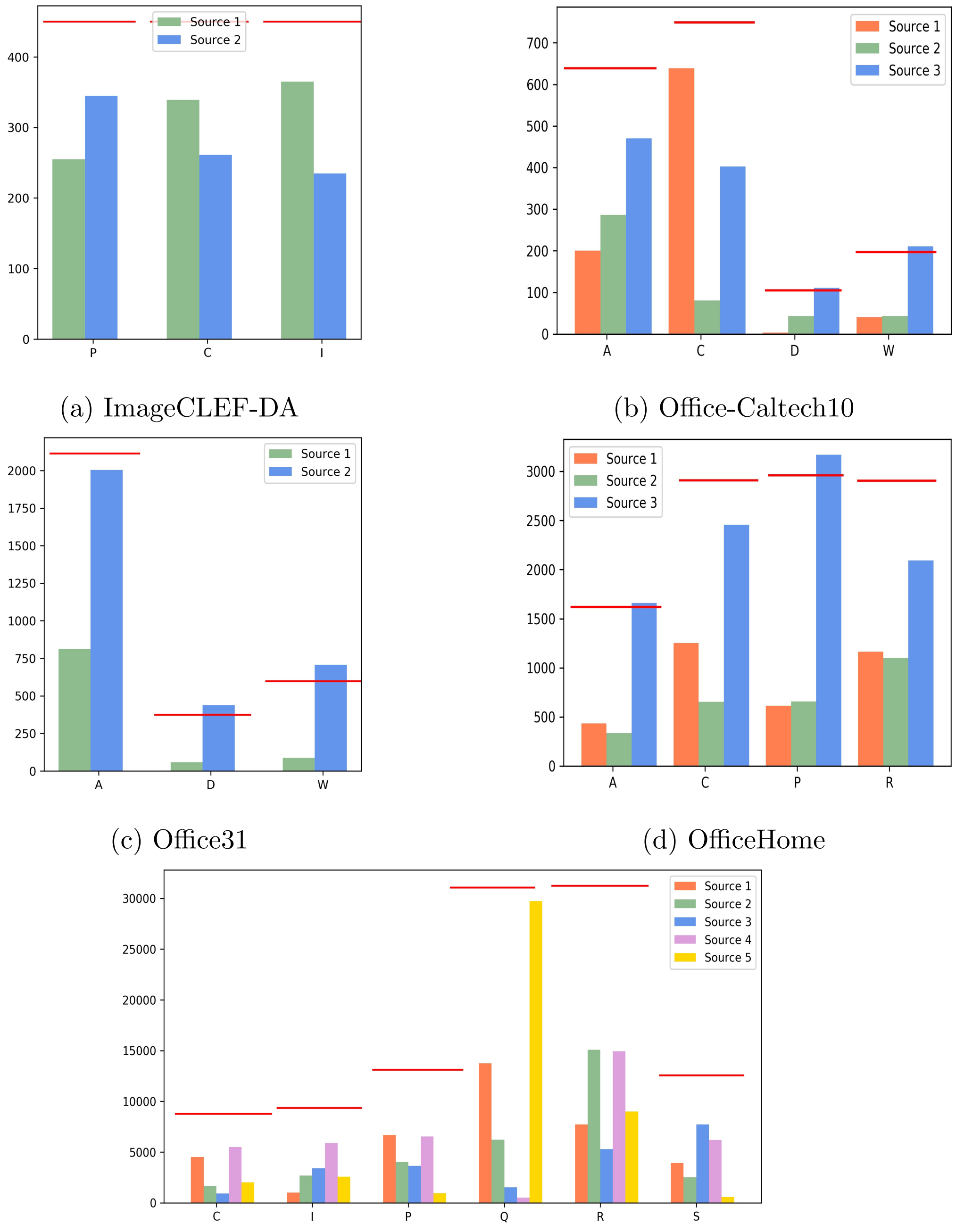
Standards	Method	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
Single best	ResNet	82.5	91.2	98.9	99.2	93.0
	MDE	89.1	94.8	99.4	**100.0**	95.8
	CyCADA	89.7	96.2	98.9	97.3	95.5
Source Combine	ResNet	87.8	86.1	99.0	98.3	92.8
	DAN	89.7	94.8	99.3	98.2	95.5
	CyCADA	91.0	95.9	99.0	97.8	95.9
Multi-Source	WBT	91.4	95.0	96.8	94.7	94.5
	MFSAN	93.8	95.1	99.1	98.7	96.7
	FADA	88.7	84.2	88.1	87.1	96.4
	MSCLDA	94.1	95.3	99.1	98.5	96.8
	CRSL	86.5	92.3	99.0	**100.0**	94.5
	DCA	94.7	**96.0**	**99.7**	99.1	97.4
	SSD	**95.0**	95.8	99.1	**100.0**	**97.5**

Table 5.5 : Accuracy (%) on dataset OfficeHome of the proposed and comparison methods

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Single best	ResNet	75.4	79.7	49.6	65.3	67.5
	DAN	75.9	80.3	56.5	68.2	70.2
	RevGard	75.8	80.4	55.9	67.9	70.0
	D-CORAL	76.3	80.3	53.6	67.0	69.3
	MRAN	77.5	**82.2**	60.0	70.4	72.5
	SAFN	81.5	77.1	57.1	70.9	71.7
	ETD	82.1	**85.7**	57.5	70.2	73.9
Source Combine	DAN	82.5	79.0	59.4	68.5	72.4
	D-CORAL	82.7	79.5	58.6	68.1	72.2
	RevGard	82.7	79.5	59.1	68.4	72.4
Multi-Source	MFSAN	80.8	79.0	60.7	70.0	72.6
	MADAN	81.5	78.2	54.9	66.8	70.4
	MIAN	80.4	79.6	63.1	69.4	73.1
	MetaMDA	**83.4**	81.2	60.5	70.2	73.8
	MSCLDA	80.6	79.9	61.4	71.6	73.4
	DCA	81.4	80.5	63.6	72.1	74.4
	SSD	83.2	81.2	**64.5**	**72.5**	**75.4**

Table 5.6 : Accuracy (%) on dataset DomainNet of the proposed and comparison methods

Standards	Method	C	I	P	Q	R	S	Avg
Single best	ResNet	39.6±0.6	8.2±0.8	33.9±0.6	11.8±0.7	41.6±0.8	23.1±0.7	26.4
	DAN	39.1±0.5	11.4±0.8	33.3±0.6	16.2±0.4	42.1±0.7	29.7±0.9	28.6
	ADDA	39.5±0.8	14.5±0.7	29.1±0.8	14.9±0.5	41.9±0.8	30.7±0.7	28.4
	MCD	42.6±0.3	19.6±0.8	42.6±1.0	3.8±0.6	50.5±0.4	33.8±0.9	32.2
Source Combine	DAN	45.4±0.5	12.8±0.9	36.2±0.6	15.3±0.4	48.6±0.7	34.0±0.5	32.1
	ADDA	47.5±0.8	11.4±0.7	36.7±0.5	14.7±0.5	49.1±0.8	33.5±0.5	32.2
	MCD	54.3±0.6	22.1±0.7	45.7±0.6	7.6±0.5	58.4±0.7	43.5±0.6	38.5
Multi-Source	M3SDA	58.6±0.5	**26.0**±0.9	52.3±0.6	6.3±0.6	62.7±0.5	49.5±0.8	42.6
	MDDA	59.4±0.6	23.8±0.8	**53.2**±0.6	12.5±0.6	61.8±0.5	48.6±0.8	43.2
	MetaMDA	62.8±0.2	21.4±0.1	50.5±0.1	15.5±0.2	64.6±0.2	50.4±0.1	44.2
	SSD	**67.2**±0.1	21.7±0.1	52.4±0.2	**20.8**±02	**67.8**±0.1	**55.3**±0.2	**47.5**



(a) ImageCLEF-DA

(b) Office-Caltech10

(c) Office31

(d) OfficeHome

(e) DomainNet

Figure 5.3 : Quantities of the target samples belonging to the source domains. Red line indicates the threshold of the dominant source domain for each target task. Source order is described in section 5.4.1.

posed combination rule in equation (5.16).

The accuracy of the proposed combination rule overtakes the greatest accuracy returned by both single source domain adaptation and the results of other combination rules. Even single source performance of SSD is superior to many baselines in Tables 5.2, 5.3, 5.4, and 5.5. It also indicates a phenomena that the transferable information is asymmetric. Taking tasks from dataset Office31 as examples, for target task $W, D \rightarrow A$, the model trained with the source domain D outperforms that trained with the source domain W. It is expected that, when learning target task $A, W \rightarrow D$, the model trained with the source domain A should perform better than that trained with the source domain W, as domains A and D show stronger connection in task $W, D \rightarrow A$. However, the fact is that the predictor from the source domain W is superior to that from the source domain A. This means that the information from domain D can be transferred to domain A, but the information from domain A might not be ideal for domain D. In other words, the transferable information between two domains is unbalanced, or one-way.

For DomainNet, the selected source domains achieve higher performance than applying all source domains using average mean and weighted average mean combinations. The proposed SSD applies three source domains, but outperforms the transfer model employing all five source domains. It indicates our method can reduce negative transfer caused by weakly-connected source domains and enhance the positive transfer. For most target tasks, the proposed combination rule can identify the dominant source domain, if it exists, and return the best performance. Even for those target tasks where the accuracy produced by the single best domain adaptation is better than that returned by the multi-source domain adaptation, $A, C, P \rightarrow R$, $A, P, R \rightarrow C$ from dataset OfficeHome, the domain discriminator can divide most target samples into the closest connected source domain. We may take as a future study the defining of a more sensitive threshold to choose the dominant source domain, and to explore the auxiliary function of other source domains when transferring knowledge across domains.

Table 5.7 : Accuracy (%) on dataset Office31 with different combination rules

Standards	A, W→D	A, D→W	W, D→A	Avg
S1	98.0	98.9	75.8	91.3
S2	**99.8**	**99.1**	**76.0**	
Mean	98.9	**99.1**	75.9	91.3
Weighted	99.3	99.0	**76.0**	91.4
Proposed	**99.8**	**99.1**	**76.0**	**91.6**

Table 5.8 : Accuracy (%) on dataset ImageCLEF-DA with different combination rules

Standards	I, C→P	I, P→C	P, C→I	Avg
S1	**79.2**	96.4	94.4	90.1
S2	**79.2**	96.4	94.7	
Mean	**79.2**	96.5	94.6	90.1
Weighted	**79.2**	**96.6**	**94.8**	**90.2**
Proposed	**79.2**	**96.6**	**94.8**	**90.2**

5.4.5 Sample Complexity Analysis

The threshold a_c for each class is defined to guarantee at least one source sample in each category is selected as a transfer sample, a measure which aims to avoid

Table 5.9 : Accuracy (%) on dataset Office-Caltech10 with different combination rules

Standards	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
S1	94.5	95.6	98.6	97.5	97.0
S2	94.9	95.8	98.6	98.5	
S3	**95.0**	**95.9**	**99.7**	**100.0**	
Mean	**95.0**	95.8	99.2	99.4	97.3
Weighted	**95.0**	95.8	99.1	98.9	97.1
Proposed	**95.0**	95.8	99.1	**100.0**	**97.5**

Table 5.10 : Accuracy (%) on dataset OfficeHome with different combination rules

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
S1	82.0	76.6	62.9	68.7	73.9
S2	81.8	78.5	63.7	70.3	
S3	**83.6**	**81.2**	**64.4**	**72.5**	
Mean	83.2	79.6	64.1	71.1	74.5
Weighted	83.2	80.6	**64.5**	71.6	74.9
Proposed	83.2	**81.2**	**64.5**	**72.5**	**75.4**

Table 5.11 : Accuracy (%) on dataset DomainNet with different combination rules. Result with symbol * is the selected source domain.

Value	C	I	P	Q	R	S	Avg
S1	64.6±0.2*	21.1±0.1	51.5±0.1*	20.9±0.2*	67.1±0.2	54.7±0.2*	46.4
S2	64.2±0.7	21.5±0.1*	51.2±0.3*	20.3±0.1*	67.1±0.2*	53.4±0.2	
S3	64.6±0.9	21.3±0.1*	50.5±0.1	20.5±0.2	67.2±0.2	54.6±0.2*	
S4	66.5±0.2*	21.3±0.2*	52.1±0.1*	20.4±0.1	66.9±0.2*	54.5±0.0*	
S5	66.8±0.1*	21.3±0.1	51.6±0.1	20.7±0.1*	67.5±0.1*	54.4±0.1	
Mean	65.6±0.8	**21.7**±0.0	51.5±0.2	20.6±0.2	67.7±0.2	54.7±0.2	47.0
Weighted	65.6±0.8	**21.7**±0.0	51.5±0.2	20.6±02	67.7±0.2	54.8±0.2	47.0
Proposed	**67.2**±0.1	**21.7**±0.1	**52.4**±0.2	**20.8**±02	**67.8**±0.1	**55.3**±0.2	**47.5**

the possibility of any unshared category. The value of a_c affects the quantity of source samples used for training. In other words, it is connected directly with the sample complexity. Taking datasets ImageCLEF-DA and OfficeHome as examples, Tables 5.12 5.13 shows the performance of source models using different quantities of training samples. $a_c = 1$ means only source samples collected by the multiple binary classifiers in equation (5.7) are used for training; $a_c = medium$ means the source samples selected using (5.7) and the threshold a_c are used for training; $a_c = 0$ means all source samples are used.

For dataset ImageCLEF-DA, which contains fewer samples and fewer categories, using the source samples selected only by the target category classifiers can achieve performances matching the high levels attained when using all source samples. Therefore, we can see for dataset OfficeHome, which has many samples as well as categories, the model trained with the half source samples achieves the highest average performance on four target tasks. When $a_c = 1$, there might be too few or even no training samples in one category, which degrades the transfer performance; when $a_c = 0$, however, there are too many inefficient source samples showing a weak connection with the target domain, which even introduces negative transfer.

Table 5.12 : Accuracy (%) on dataset ImageCLEF-DA with different quantities of the transfer source samples

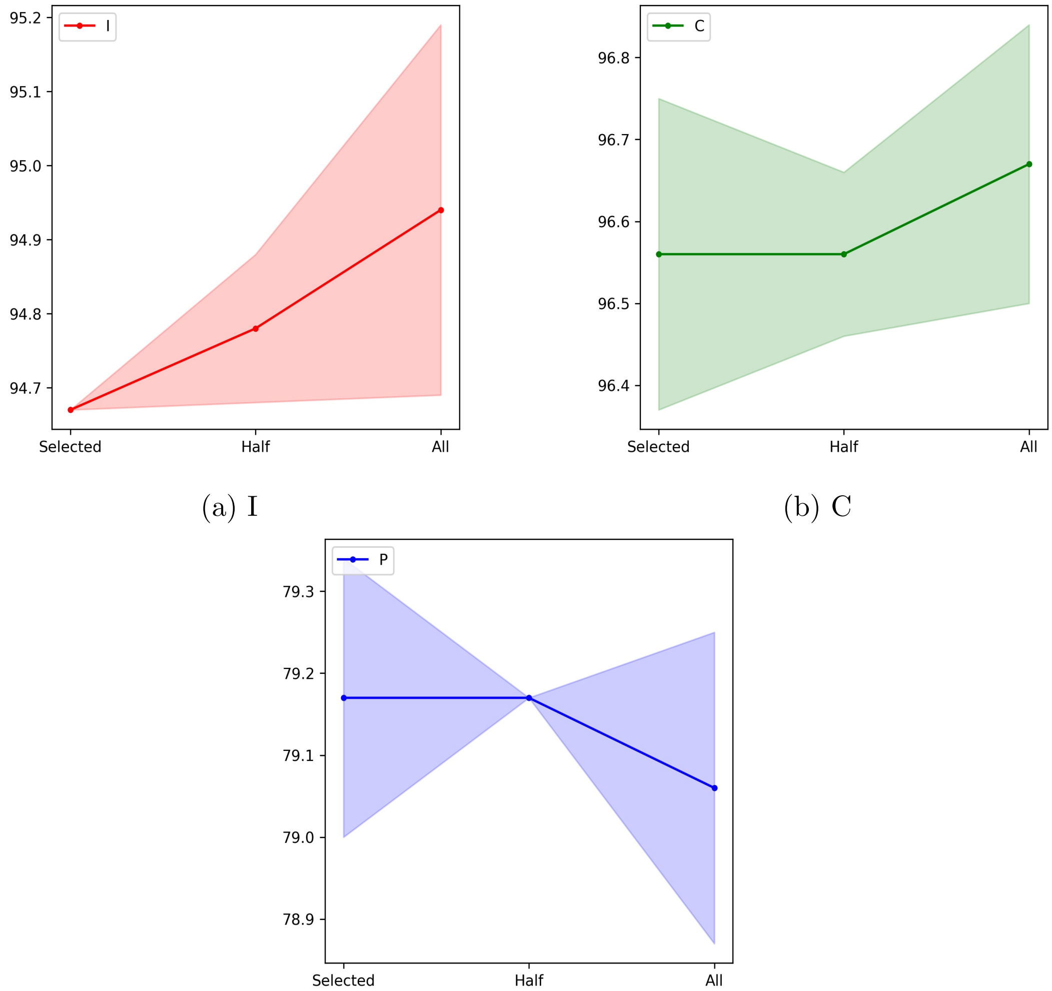
Value	I, C→P	I, P→C	P, C→I	Avg
a_c=1	**79.2**	96.6	94.7	**90.2**
a_c=medium	**79.2**	96.6	94.8	**90.2**
a_c=0	79.1	**96.7**	**94.9**	**90.2**

Table 5.13 : Accuracy (%) on dataset OfficeHome with different quantities of the transfer source samples

Value	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
a_c=1	81.2	79.2	62.2	70.6	73.3
a_c=medium	**83.2**	**81.2**	**64.5**	**72.5**	**75.4**
a_c=0	82.0	**81.2**	62.9	**72.5**	74.7

Figs. 5.4 and 5.5 show the accuracy of sample complexity with the standard deviation. “Without a_c” indicates $a_c = 1$, where only source samples predicted correctly by the category classifier are used without considering if there is no sample in some classes. “Proposed” indicates the proposed selective strategy with threshold a_c = medium, and “All” indicates $a_c = 0$, where all source samples are used. It can be seen that the models trained with the distilled source domains with threshold

a_c perform with greater stablility than those trained with all source samples, and achieves the best performance on most tasks.



(a) I

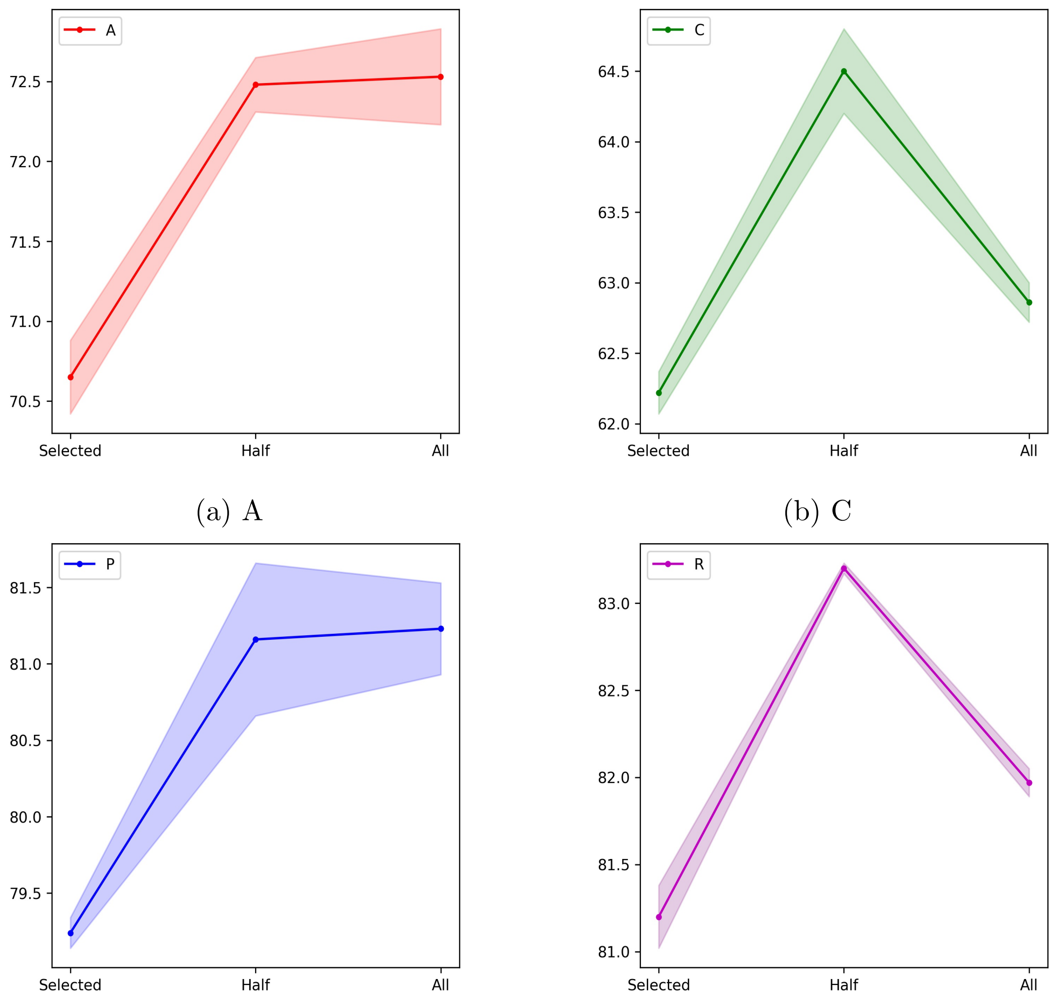
(b) C

(c) P

Figure 5.4 : Accuracy (%) on dataset ImageCLEF-DA with different quantities of the transfer source samples.

5.4.6 Ablation study

Tables 5.14 and 5.15 show the ablation study of the proposed method on datasets ImageCLEF-DA and OfficeHome. The influence of the domain adaptation loss on the domain-level L_d, domain adaptation loss on the class-level L_c, pseudo-labeled



(a) A

(b) C

(c) P

(d) R

Figure 5.5 : Accuracy (%) on dataset OfficeHome with different quantities of the transfer source samples.

constraint L_{supt}, cross-entropy loss on source domain L and cross-domain loss L_{cro} is validated by moving each of them when training.

The proposed method achieves the highest transfer performance. The cross-entropy loss L on the source domain is the most essential factor for training a well performed transfer model, since without it, the performance decreases significantly. Model trained with constraint L_{supt} based on the pseudo-labeled target domain gains higher performance than that trained without, meaning it plays an important role in training the model compared with other loss functions except for the source cross-entropy loss L_{cro}. Domain adaptation loss on the domain-level L_d and that on the class-level L_c, and the cross-domain constraint L_{cro} contribute as much as each other to the transfer model training.

Table 5.14 : Accuracy (%) of the ablation study on dataset ImageCLEF-DA

Constraint	I, C→P	I, P→C	P, C→I	Avg
L_{supt}	79.3	96.4	94.4	90.0
L	79.2	96.4	94.4	90.0
L_d	79.3	96.4	94.7	90.1
L_{cro}	**79.5**	96.3	94.7	**90.2**
L_c	79.3	96.5	94.7	**90.2**
Proposed	79.2	**96.6**	**94.8**	**90.2**

5.4.7 Application of Sample and Source Distillation in Existing Methods

Our strategy can be applied to existing domain adaptation methods, whereby learning a small scale of parameters with the selected source samples, the performance on the target domain can be improved. Table. 5.16 shows the performance

Table 5.15 : Accuracy (%) of the ablation study on dataset OfficeHome

Constraint	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
L_{supt}	79.1	**81.9**	62.1	**72.5**	73.9
L	80.8	79.3	61.9	70.0	73.0
L_d	82.0	81.1	63.4	72.2	74.7
L_{cro}	81.7	81.3	63.2	72.3	74.6
L_c	82.1	81.0	63.8	72.4	74.8
Proposed	**83.2**	81.2	**64.5**	**72.5**	**75.4**

of applying the proposed method to some existing domain adaptation methods. Dynamic adversarial adaptation network (DAAN) (Yu et al., 2019a) learns the relationship between the marginal and conditional distributions dynamically, and matches the domains based on the adversarial metric. MRAN (Zhu et al., 2019b) employs multi-representation to collect richer transferable knowledge, and extends MMD to adapt conditional distributions. MFSAN (Zhu et al., 2019a) is a multi-source domain adaptation network aligning both distributions and classifiers based on MMD. By adding the transfer samples selection strategy and re-training the model using self-training, the performance has been improved beyond the original algorithms.

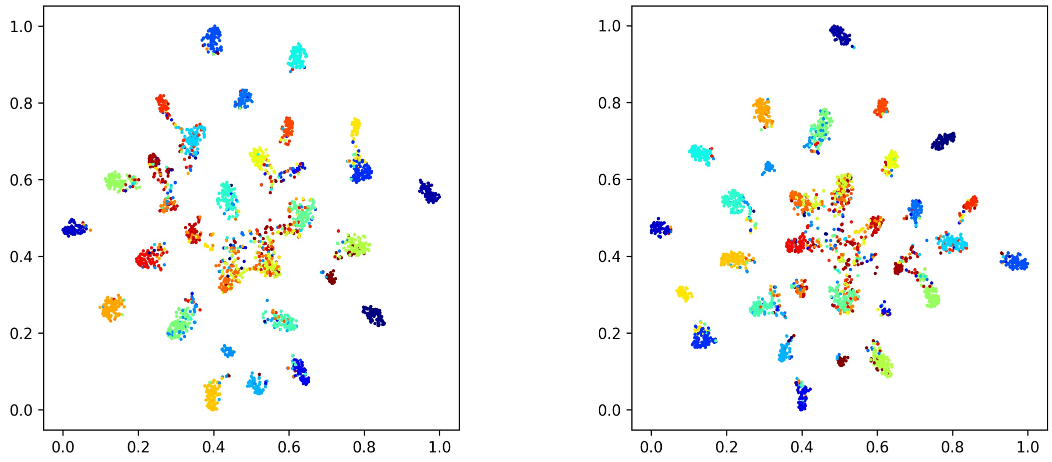
5.4.8 Visualization Analysis

To better show the transferable ability between source and target domains, taking target tasks $I, C \rightarrow P$ from dataset ImageCLEF-DA and $A, D, W \rightarrow C$ from dataset Office-Caltech10 as examples for two-source and three-source domain adaptation, Figs. 5.6 and 5.7 show the T-SNE visualization of the corresponding target domains, respectively.

T-SNE provides a direct observation for the classification ability of different

Table 5.16 : Accuracy (%) on dataset OfficeHome of existing methods with transfer sample selection

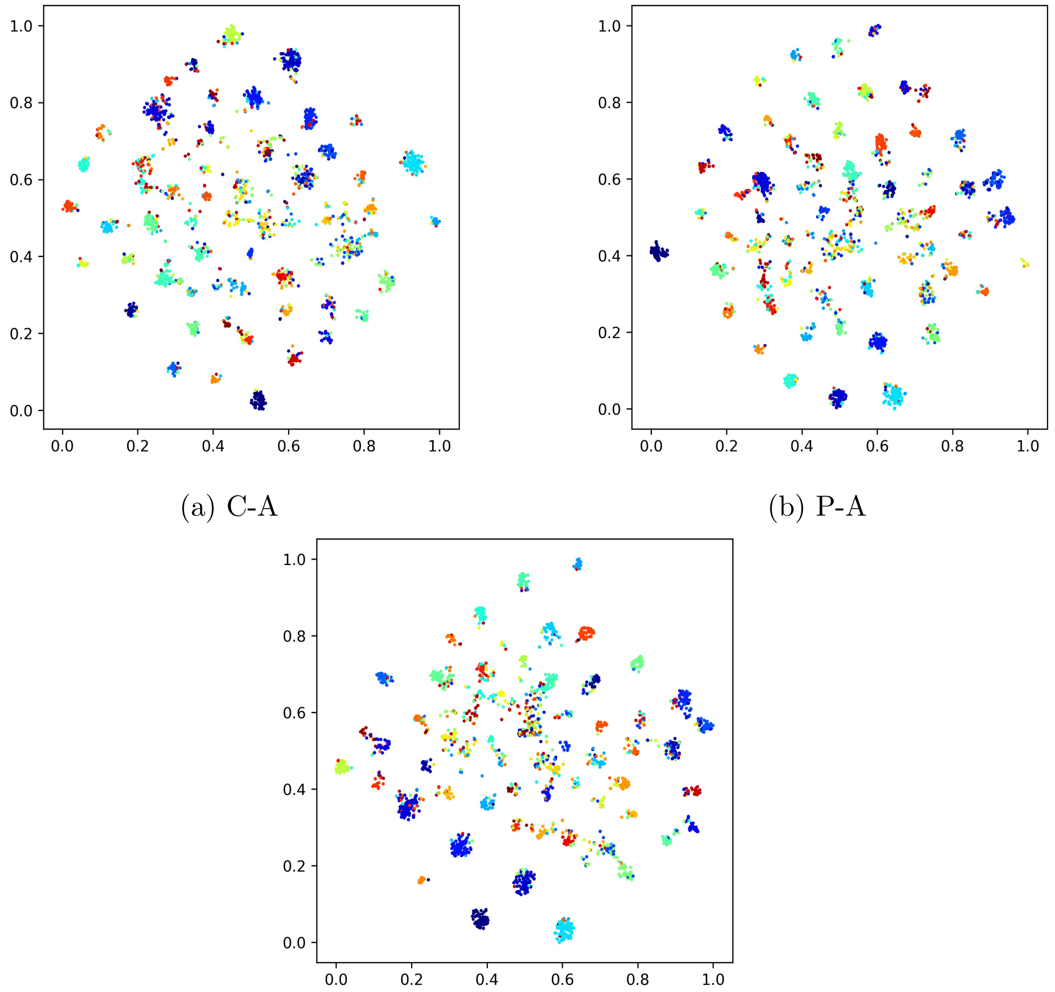
Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Single best	DAAN	74.2	78.5	53.1	64.8	67.7
	DAAN+SSD	76.3	79.5	54.5	65.6	69.0
	MRAN	77.5	82.2	60.0	70.4	72.5
	MRAN+SSD	80.3	**83.2**	61.8	72.0	74.3
Multi-Source	MFSAN	80.8	79.0	60.7	70.0	72.6
	MFSAN+SSD	82.2	81.1	62.8	**72.4**	74.6
	MSCLDA	80.6	79.9	61.4	71.6	73.4
	MSCLDA+SSD	**82.3**	81.2	**62.9**	**72.4**	**74.7**



(a) W-A

(b) D-A

Figure 5.6 : T-SNE visualization of target domain A from dataset Office-31.



(a) C-A

(b) P-A

(c) R-A

Figure 5.7 : T-SNE visualization of target domain A from OfficeHome.

source domains, and indicates misalignment samples. The source classifier trained with samples from domain I separates the target samples more clearly with a large distance compared with the source classifier from domain C, although the accuracy of two classifiers is the same. The same situation can be found for target domain W from dataset Office-Caltech10, as source classifier D groups samples with both quite short intra-class distance and large inter-class distance. It is obvious that different source domains perform differently on the target domain, making the choice of the most suitable source domain and defining its dominating position during training absolutely are essential for enhancing the performance.

5.5 Summary

This chapter proposes a transfer sample and source distillation method. Compared with many existing methods, instead of measuring the distance between the source and target samples, we build a series of binary classifiers based on the similarity of a sample belonging to a category to divide transfer and inefficient samples. A two-step selective strategy is developed to filter out inefficient samples and the source domains. According to the results of our experiment, a reduction of training samples does not always degrade classification accuracy. On the contrary, a classifier learned with the selected transfer samples can improve accuracy by eliminating information that might confuse the classifier.

In the future, measuring the relationships and exploring the asymmetric transfer information between source and target domains will be valuable topics to pursue. These investigations could possibly lead to important discoveries in the mechanism of transferable information and their influences on a broad range of different data.

Chapter 6

Multi-Source-Free Domain Adaptation with Generally Auxiliary Model Training

6.1 Introduction

As mentioned before, most existing domain adaptation methods require the access to source data without considering the privacy issue. Data privacy and security attract attention in many situations and applications. To handle privacy concerns in domain adaptation, source data free methods are proposed (Liang et al., 2020; Agarwal et al., 2021; Hou and Zheng, 2020). Two techniques are widely used when transferring source knowledge to the target domain without source data- data generation (Li et al., 2020d) and pseudo-labeling (Kim et al., 2020). A recent method-progressive graph learning- adapts source model to the target domain containing unshared label subset without matching data distributions (Luo et al., 2022). The original hypothesis space is divided into shared label space and unknown label space, where the source and target risks are minimized by learning a tighter error bound. sample-level and manifold-level shifts are filled by replacing the source label with the pseudo labels gradually using an adversarial training strategy.

Data generation methods generate fake samples from source classes based on the pre-trained source model, and adapt the generated samples to the target domain to achieve distribution adaptation. Domain impression builds a generative framework to deal with source-free domain adaptation with noise (Kurmi et al., 2021b). It includes generation module and adaptation module. Generation module first obtains samples which can be divided correctly by the source classifier to train a discriminator, then the adaptation module fits the source classifier to the target domain by minimizing the likehood loss using an adversarial way.

Since data generation methods often generate extra fake source samples which require more computer memory, to avoid this problem, pseudo-labeling learned from target domain becomes popular recently. Pseudo-labeling can provide pseudo target labels to help fit the source model to the target domain under the supervision of pseudo labels. Generalized source-free domain adaptation proposes local structure clustering to divide target samples into multiple groups by finding the semantically similar neighbors (Yang et al., 2021c). Spare domain attention layer is applied to the source model to ensure its performance on target domain and protect the performance on source domain at the same time. And continual domain adaptation without source data is extended to deal with target domains in sequence.

Aforementioned source-free methods focus on single source domain adaptation, but rarely consider the situation of multiple source domains. In addition, they lack enough exploration of the influence of the imbalanced data in domain adaptation. In this chapter, we propose a multi-source data-free domain adaptation method with generally auxiliary model training (GAM). For each source domain, the method constructs a generally auxiliary task from other source domains to improve the generality of the source model when performing on a new domain, and protect the data privacy of other source domains by sharing only source parameters. Furthermore, we introduce coefficients based on the number of samples to solve the class imbalance problem. The contributions of this chapter are summarized as follows.

- We propose a multi-source data-free domain adaptation method which is rarely explored in previous studies. Both specific and general source models are learned to provide across-domain ability to perform on a new domain.

- We develop a general model based on auxiliary training that can fit multiple domains by sharing source parameters, which can improve the generality of source models to provide initial pseudo target labels with high quality, and concurrently protect data privacy.

- We introduce class balance coefficients to source-free domain adaptation, which is ignored in previous studies. It can eliminate the influence of class imbalance and improve the performance of learned classifiers.

6.2 Problem Setting and Notations

In this chapter, we deal with unsupervised data-free domain adaptation with multiple sources under closed set, and extend the method to partial and open-set domain adaptation in experiment. Notations used in this chapter are described in Table 6.1.

Table 6.1 : Notations and descriptions.

Notation	Description
$\mathcal{D}_{s_k}$, $\mathcal{D}_t$	source/target domain, k is source index
n_{s_k}, n_t	number of samples from source/target domain
$n_{s_k}^c$	number of samples in cth category
$\boldsymbol{x}_{s_k}$, $\boldsymbol{x}_t$	sample from the source/target domain
$\boldsymbol{y}_{s_k}$	corresponding label of $\boldsymbol{x}_{s_k}$
$\tilde{\boldsymbol{y}}_{s_k}$	corresponding smooth label of $\boldsymbol{x}_{s_k}$
$B_{\boldsymbol{y}_{s_k}}$	class balance coefficient of $\boldsymbol{y}_{s_k}$
P_{s_k}	private classifier of the kth source domain
P_g	generally auxiliary classifier of source domains
$\boldsymbol{\omega}$	combination parameter for generating generally auxiliary classifier
v_t^c	deep clustering center of the cth class from target domain

6.3 The Proposed Generally Auxiliary Model Method under Closed Set

The proposed method includes specific source model training, generally auxiliary model training, pseudo label collecting and target task predicting. Specific source model training learns specific multiple source models. Generally auxiliary model training constructs a general model from multiple source domains, which we anticipate will improve the generality of source models when performing on a new domain. Pseudo label collecting provides pseudo target labels using a deep clustering method, which is used to supervise the model re-training when fitting the source models to the target domain. The re-trained source models finally predict the target task. The whole framework is shown in Fig. 6.1, the top figure (a) displays the training of source models, including specific source model training and generally auxiliary model training; the bottom figure (b) indicates the process of fitting the source models to the target domain, including pseudo label collecting and target task predicting.

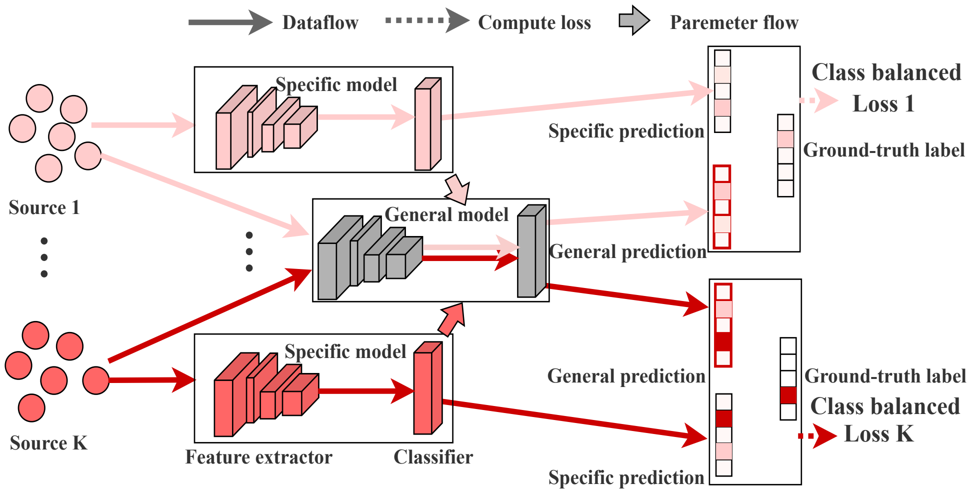
6.3.1 Specific Source Model Pre-training

In this section, we introduce the training of specific source models. For the kth source domain, based on the structural risk minimization principle (Vapnik and Vapnik, 1998), the error between the predictions of the classifier and the ground-truth labels is minimized to learn the classifier, which is expressed as:

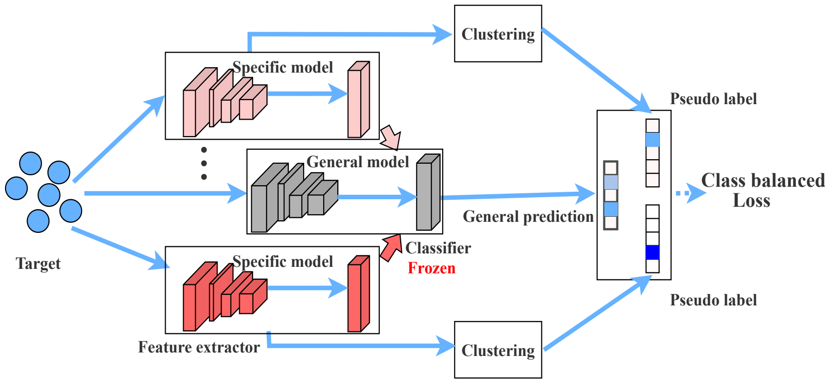
$$P_{s_k} = \underset{\substack{P_{s_k} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}_{s_k}}}{\arg\min} \; L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{y}_{s_k}), \tag{6.1}$$

ϕ indicates the pre-trained backbone, ϕ_k is the specific feature extractor for each source domain. L indicates cross-entropy loss, which is:

$$L = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}^i)))). \tag{6.2}$$



(a) Source model pre-training: specific source model training and generally auxiliary model training.



(b) Model adapting: pseudo label collecting and target task predicting.

Figure 6.1 : The procedure of the proposed method.

To avoid over-confidence within the network and improve the learning speed of multi-class classifiers, we apply label smoothing to transform the labels from hard to soft (Müller et al., 2019; Liang et al., 2020). For target label $\boldsymbol{y}_{s_k}$, a one-hot vector where the value of its correct class equals 1 while the others equal 0, label smoothing turns the original label to:

$$\tilde{\boldsymbol{y}}_{s_k} = (1-\mu)\boldsymbol{y}_{s_k} + \mu/C, \tag{6.3}$$

where μ is smoothing parameter, C is the number of classes. Equation (6.2) can be re-written as:

$$L = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \tilde{\boldsymbol{y}}_{s_k}^i \log(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}^i)))). \tag{6.4}$$

Considering the imbalanced data distribution, especially in a real-world dataset, we introduce the class balance coefficient to overcome the data under-representation problem caused by imbalanced data, which can affect the performance of the classifier on any classes containing fewer samples (Cui et al., 2019). For a source sample in the cth class, the class balance coefficient is:

$$B_{\boldsymbol{y}_{s_k}} = \frac{1-\xi}{1-\xi^{n_{s_k}^c}}, \tag{6.5}$$

ξ is balance parameter. Loss function function in equation (6.4) is re-formulated as:

$$L = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} B_{\boldsymbol{y}_{s_k}^i} \tilde{\boldsymbol{y}}_{s_k}^i \log(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}^i)))). \tag{6.6}$$

Private specific source classifier in equation (6.1) can be re-written as:

$$P_{s_k} = \underset{\substack{P_{s_k} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}_{s_k}}}{\arg\min} \; L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}, B_{\boldsymbol{y}_{s_k}}). \tag{6.7}$$

6.3.2 Generally Auxiliary Model Training

Generally, where multiple source domains share the same classes, richer information is beneficial to the classifier learning. As our final purpose is predicting a new unlabeled target domain, one that is different from all source domains, it encourages us to learn a model that has better generality than specific source domains. An entirely new and independent model, however, will introduce more parameters. So, to avoid this, we build a general model based on all specific source models to ensure that the learned source models can be performed on multiple domains. Considering the data privacy, for each source domain, only model parameters alone are shared, without access to the data from other source domains. The general source model is expressed as:

$$P_g = G(P_{s_k}, \boldsymbol{\omega}) = \sum_{k=1}^{K} \omega_k P_{s_k}, \tag{6.8}$$

$\boldsymbol{\omega}$ is combination parameter learned automatically in the training.

The general model acts as an auxiliary task for each source domain to improve the generality of specific source models, for the kth source domain, the general model is parameterized by:

$$P_g = \underset{\substack{P_g \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}_{s_k}}}{\arg\min} \ L(P_g(\Phi(\boldsymbol{x}_{s_k})), \tilde{\boldsymbol{y}}_{s_k}, B_{\boldsymbol{y}_{s_k}}), \tag{6.9}$$

where $P_g(\Phi(\boldsymbol{x}_{s_k})) = G(P_{s_{k'}}(\phi_{k'}(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{\omega}), k' = 1, \cdots, K$, L is defined in equation (6.6).

The total loss function of P_{s_k} is:

$$\begin{aligned} P_{s_k} = \underset{\substack{P_{s_k} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}_{s_k}}}{\arg\min} \ & L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}, B_{\boldsymbol{y}_{s_k}}) + \\ & L(P_g(\Phi(\boldsymbol{x}_{s_k})), \tilde{\boldsymbol{y}}_{s_k}, B_{\boldsymbol{y}_{s_k}}). \end{aligned} \tag{6.10}$$

6.3.3 Pseudo Label Collecting and Target Task Predicting

When tackling the target task, only pre-trained source models are available, since traditional domain adaptation relying on data matching essentially fails to handle this setting. To fit the source models to the target domain, we expect to transform the target data into a latent feature space similar to the corresponding source feature space where the classifier is trained. Since the target domain is unlabeled, one method is to generate pseudo labels to supervise the data transformation. A self-supervised clustering strategy is used to pseudo-label the target samples (Liang et al., 2020; Caron et al., 2018). For the cth class in the kth source domain, the initial clustering center can be expressed as:

$$\boldsymbol{v}_t^c = \frac{\sum_{i=1}^{\hat{n}_t^c} P_g(\Phi(\boldsymbol{x}_t^i)) \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n}_t^c} P_g(\Phi(\boldsymbol{x}_t^i))}, \tag{6.11}$$

$\hat{n}_t^c$ is the number of samples in the cth class predicted by the classifier P_g. The initial pseudo label of sample $\boldsymbol{x}_t$ is:

$$\hat{\boldsymbol{y}}_t = \arg\min_c \mathrm{Dis}(\phi_k(\phi(\boldsymbol{x}_t)), \boldsymbol{v}_t), \boldsymbol{v}_t = [\boldsymbol{v}_t^1, \cdots, \boldsymbol{v}_t^C], \tag{6.12}$$

Dis indicates cosine distance, which has the advantage of learning the similarity between features even their geometric distance is far.

Updating the initial centers using the pseudo labels obtained in equation (6.12), the new cluster center of the cth class and pseudo label can be expressed as:

$$\begin{aligned} \boldsymbol{v}_t^c &= \frac{\sum_{i=1}^{\hat{n'}_t^c} \mathbb{1}_{\hat{\boldsymbol{y}}_t^i = c} \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n'}_t^c} \mathbb{1}_{\hat{\boldsymbol{y}}_t^i = c}}, \\ \hat{\boldsymbol{y}}_t &= \arg\min_c \mathrm{Dis}(\phi_k(\phi(\boldsymbol{x}_t)), \boldsymbol{v}_t), \\ \boldsymbol{v}_t &= [\boldsymbol{v}_t^1, \cdots, \boldsymbol{v}_t^C], \end{aligned} \tag{6.13}$$

$\hat{n'}_t^c$ is the number of samples in the cth class predicted by clustering.

After collecting the target pseudo labels, the training process of fitting the source models is achieved by reducing the error between the outputs of the general model and the pseudo labels, which is formulated as:

$$P_g = \arg\min_{\substack{\phi, \phi_k \\ x_t \sim \mathcal{D}_t}} L(P_g(\Phi(\boldsymbol{x}_t)), \hat{\boldsymbol{y}}_t, B_{\hat{\boldsymbol{y}}_t^i}), \tag{6.14}$$

where entropy loss L is:

$$\begin{aligned} L &= -\frac{1}{n_t}\sum_{i=1}^{n_t} B_{\hat{\boldsymbol{y}}_t^i}\hat{\boldsymbol{y}}_t^i \log(P_g(\Phi(\boldsymbol{x}_t^i))), \\ B_{\hat{\boldsymbol{y}}_t^i} &= \frac{1-\xi}{1-\xi^{\hat{n'}_t^c}}. \end{aligned} \tag{6.15}$$

the parameters of P_g are frozen, and feature extraction network Φ is re-trained.

To balance the large domain gap which may harm the transfer, information maximization loss which parameterizes the target outputs being individually certain and globally diverse is employed to control the target outputs as with the one-hot vector (Hu et al., 2017; Liang et al., 2020).

$$L_{div} = \sum \bar{\boldsymbol{p}}_t \log(\bar{\boldsymbol{p}}_t), \tag{6.16}$$

$\bar{\boldsymbol{p}}_t = \frac{1}{n_t}\sum_{i=1}^{n_t} P_g(\Phi(\boldsymbol{x}_t^i))$ is a C-dimension vector. The re-training process is parameterized by:

$$P_g = \underset{\substack{\Phi \\ x_t \sim \mathcal{D}_t}}{\arg\min}\ L(P_g(\Phi(\boldsymbol{x}_t)), \hat{\boldsymbol{y}}_t, B_{\hat{\boldsymbol{y}}_t^i}) + L_{div}. \tag{6.17}$$

The target label is:

$$\boldsymbol{y}_t = P_g(\Phi(\boldsymbol{x}_t)). \tag{6.18}$$

The whole algorithm is summarized in Algorithms 6 and 7

Algorithm 6 Multi-source-free domain adaptation: Pre-training

```
Input: Source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$.
Initialize: Feature extractors $\phi$ and $\phi_k$, specific source classifier $P_{s_k}$.
for $\epsilon = 1$, $\epsilon < \mathcal{I}_s$, $\epsilon++$, do
    Transform the source labels into smoothing labels as in (6.3),
    Calculate class balanced coefficient as in (6.5);
    Construct generally auxiliary source model as in (6.8);
    Train specific source model and general model as in (6.10).
end for
Output: Specific source model and general model.
```

Algorithm 7 Multi-source-free domain adaptation: Adapting

```
Input: Target domain $\mathcal{D}_t$.
Pre-trained: Specific source model and general model.
for $\epsilon = 1$, $\epsilon < \mathcal{I}_t$, $\epsilon++$, do
    Calculate initial cluster centers and pseudo labels as in (6.11), (6.12),
    Update cluster centers and pseudo target labels as in (6.13);
    Compute balance coefficient of target domain as in (6.15);
    Compute information maximization loss as in (6.16);
    Re-train source model by freezing classifier layers as in (6.17).
end for
Predict target label as in (6.18).
Output: Target label $\boldsymbol{y}_t$.
```

6.4 The Proposed Generally Auxiliary Model Method under Closed Set with Sample and Source Distillation

As discussed in chapter 5, it is the sample quality not only quantity that affects the performance of transferring. To enhance the transfer ability, the two-step selective strategy, which distills both source inefficient samples and domains, is adopted to improve the proposed generally auxiliary model training method.

To select source samples, target pseudo labels with high confidence being correct are first collected. The collection is expressed as:

$$\hat{\boldsymbol{y}}_t = \wedge P_g(\Phi(\boldsymbol{x}_t)), \tag{6.19}$$

$\wedge$ is the operation to select target pseudo labels whose maximum probability value returned by P_g is higher than the median value of the target samples from the same class predicted by P_g. After collection target pseudo labels, category classifier P_c is trained to identify a sample from its corresponding class, which is:

$$L_{bce} = \sum_{c=1}^{C} L(P_c(\phi(\boldsymbol{x}_t)), I(\hat{\boldsymbol{y}}_t, c)). \tag{6.20}$$

where $I(\hat{\boldsymbol{y}}_t, c) = 1_{\hat{\boldsymbol{y}}_t = c}$, c indicates the cth class.

Apply source samples to $\{P_c\}_{c=1}^{C}$, denote the prediction as:

$$I(\hat{\boldsymbol{y}}_{s_k}, c) = P_c(\phi(x_{s_k})). \tag{6.21}$$

If $I(\hat{\boldsymbol{y}}_{s_k}, c) = 1$ with the highest probability value, denote $\hat{\boldsymbol{y}}_{s_k} = c$, if the ground-truth label $\boldsymbol{y}_{s_k} = c$, the source sample is kept for further training.

Since the generally auxiliary classifier P_g is constructed from specific source classifiers $\{P_{s_k}\}_{k=1}^{K}$ with combination vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \cdots, \omega_K]$, if the kth source weight ω_k gains very large value compared with other source domain weights, the corresponding source domain is regarded as dominant source domain. Here we define the threshold as:

$$\begin{aligned} &\text{Dominant is true}: \\ &\text{if}: \omega_k - (1-\omega_k) > \frac{1}{K}. \end{aligned} \tag{6.22}$$

Denote the distilled source domains as $\{\mathcal{D}'\}_{k=1}^{K}$, the private source specific and general models in equation (6.10) are re-trained using the selected source samples as:

$$\begin{aligned} P_{s_k} = \mathop{\arg\min}_{\substack{P_{s_k} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}'{s_k}}} \quad & L(P_{s_k}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}, B_{\boldsymbol{y}_{s_k}}) + \\ & L(P_g(\Phi(\boldsymbol{x}_{s_k})), \tilde{\boldsymbol{y}}_{s_k}, B_{\boldsymbol{y}_{s_k}}), \end{aligned} \tag{6.23}$$

Adapt the re-trained source generally auxiliary model to target domain as in section 6.3.3, the target task can be completed.

The whole algorithm is summarized in Algorithms 8

Algorithm 8 Multi-source-free domain adaptation with source and sample distillation.

```
Input: Target domain $\mathcal{D}_t$.
Pre-trained: Specific source model and general model.
Collect target pseudo labels as in (6.19);
Train target category classifier as in (6.20);
Distill source samples as in (6.21);
Select dominant source domain as in (6.22);
for $\epsilon = 1$, $\epsilon < \mathcal{I}_s$, $\epsilon + +$, do
    Re-train source specific and general models based on distilled source domains as in (6.23).
end for
Input source general model based on distilled source domains;
for $\epsilon = 1$, $\epsilon < \mathcal{I}_t$, $\epsilon + +$, do
    Calculate initial cluster centers and pseudo labels as in (6.11), (6.12);
    Update cluster centers and pseudo target labels as in (6.13);
    Compute balance coefficient of target domain as in (6.15);
    Compute information maximization loss as in (6.16);
    Re-train source model by freezing classifier layers as in (6.17).
end for
Predict target label as in (6.18).
Output: Target label $\boldsymbol{y}_t$.
```

6.5 The Proposed Generally Auxiliary Model Method under Partial Set

In partial domain adaptation, given the pre-trained source private specific and general models using equation (6.10) in section 6.3.2, the main problem is how to reduce the effect of unshared source classes. When initializing clustering centers and pseudo labels of target domain as in equations (6.11) and (6.12), ideally, the unshared source classes gain very small probability values in target domain, which result in the low probability of the target samples belonging to the unshared source classes. Based on this assumption, we remove the target clustering centers which contain none samples during training iteratively to eliminate the negative influence of outlier source classes. Denote source label space as $\mathcal{C}$ containing C classes, target label space as $\mathcal{C}_t$ where $\mathcal{C}_t \subseteq \mathcal{C}$, this operation can be expressed as:

$$\begin{aligned} \boldsymbol{N} &= \text{Count}(\text{Rank}(P_g(\Phi(\boldsymbol{x}_t)))), \\ \boldsymbol{N} &= [n^1, \cdots, n^C], c \in \mathcal{C}. \end{aligned} \tag{6.24}$$

Rank means the operation to rank the probability values returned by classifier P_g, Count means the operation to count the number of a class gaining the maximum value. If $n^c > 0$, corresponding class c is added to target label space $\mathcal{C}_t$, otherwise, we remove the corresponding class from corresponding source domain.

The initial target clustering centers and pseudo labels in equations (6.11) and (6.12) are expressed as:

$$\boldsymbol{v}_t^c = \frac{\sum_{i=1}^{n^c} P_{s_k}(\phi_k(\phi(\boldsymbol{x}_t^i))) \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{n^c} P_{s_k}(\phi_k(\phi(\boldsymbol{x}_t^i)))}, c \in \mathcal{C}_t \tag{6.25}$$

and

$$\hat{\boldsymbol{y}}_t = \arg\min_c \text{Dis}(\phi_k(\phi(\boldsymbol{x}_t), \boldsymbol{v}_t)), \boldsymbol{v}_t = \{\boldsymbol{v}_t^c\}_{c \in \mathcal{C}_t}, \tag{6.26}$$

n^c is the number of samples in the cth class computed using equation (6.24).

The update in equation (6.13) can be re-written as:

$$\begin{aligned} \boldsymbol{v}_t^c &= \frac{\sum_{i=1}^{\hat{n}^c} \mathbb{1}_{\hat{\boldsymbol{y}}_t^i = c} \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n}^c} \mathbb{1}_{\hat{\boldsymbol{y}}_t^i = c}}, c \in \mathcal{C}_t \\ \hat{\boldsymbol{y}}_t &= \arg\min_c \mathrm{Dis}(\phi_k(\phi(\boldsymbol{x}_t), \boldsymbol{v}_t)), \\ \boldsymbol{v}_t &= \{\boldsymbol{v}_t^c\}_{c \in \mathcal{C}_t}, \end{aligned} \tag{6.27}$$

$\hat{n}^c$ is the number of samples in the cth class predicted by clustering.

Entropy loss and balance coefficient in equation (6.15) is:

$$\begin{aligned} L &= -\frac{1}{n_t} \sum_{i=1}^{n_t} B_{\hat{\boldsymbol{y}}_t^i} \hat{\boldsymbol{y}}_t^i \log(P_g(\Phi(\boldsymbol{x}_t^i))), \\ B_{\hat{\boldsymbol{y}}_t^i} &= \frac{1-\xi}{1-\xi^{\hat{n}^c}}, c \in \mathcal{C}_t. \end{aligned} \tag{6.28}$$

The whole algorithm is summarized in Algorithms 9

Algorithm 9 Multi-source-free domain adaptation under partial set.

```
Input: Target domain $\mathcal{D}_t$.
Pre-trained: Specific source model and general model.
Predict target class number $\boldsymbol{N}$ as in (6.24);
Select target label if $n^c > 0, n^c \in \boldsymbol{N}$;
for $\epsilon = 1$, $\epsilon < \mathcal{I}_t$, $\epsilon + +$, do
    Calculate initial cluster centers and pseudo labels as in (6.25), (6.26);
    Update cluster centers and pseudo target labels as in (6.27);
    Compute balance coefficient of target domain as in (6.28);
    Compute information maximization loss as in (6.16);
    Re-train source model by freezing classifier layers as in (6.17).
end for
Predict target label as in (6.18).
Output: Target label $\boldsymbol{y}_t$.
```

6.6 The Proposed Generally Auxiliary Model Method under Open-Set

In open-set domain adaptation, the main problem is how to identify unknown target classes and divide shared classes simultaneously based on the given pre-trained source private specific and general models. Denote source label space as $\mathcal{C}_s$ containing C_s classes, target label space as $\mathcal{C}$ containing C classes, where $\mathcal{C}_s \subseteq \mathcal{C}$, the unknown target label space $\mathcal{C} \backslash \mathcal{C}_s$ is denoted as class $C_s + 1$ in the source domain. Given P_{s_k} and P_g trained using equation (6.10) in section 6.3.2, generally, the unknown target samples can gain lower probability values than the known samples. To determine unknown samples from the known, a threshold is defined to divide samples, which is:

$$a_o = \frac{\sum -P_g(\Phi(\boldsymbol{x}_t)) \log(P_g(\Phi(\boldsymbol{x}_t)))}{\log(C_s)}. \tag{6.29}$$

If the maximum probability value of a target sample is higher than a_o, we regard it as a sample from known classes. Otherwise, we regard it as unknown classes with label C_s+1, and these unknown samples are not used to calculate clustering centers.

The initial target clustering centers and pseudo labels in equations (6.11) and (6.12) are re-written as:

$$\boldsymbol{v}_t^c = \frac{\sum_{i=1}^{\hat{n}_t^c} P_{s_k}(\phi_k(\phi(\boldsymbol{x}_t^i))) \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n}_t^c} P_{s_k}(\phi_k(\phi(\boldsymbol{x}_t^i)))}, c \in \mathcal{C}_s \tag{6.30}$$

and

$$\hat{\boldsymbol{y}}_t = \arg\min_c \operatorname{Dis}(\phi_k(\phi(\boldsymbol{x}_t), \boldsymbol{v}_t)), \boldsymbol{v}_t = \{\boldsymbol{v}_t^c\}_{c \in \mathcal{C}_s}, \tag{6.31}$$

$\hat{n}_t^c$ is the number of samples in the cth class computed by classifier P_{s_k}.

The update in equation (6.13) can be re-written as:

$$\begin{aligned}
\boldsymbol{v}_t^c &= \frac{\sum_{i=1}^{\hat{n'}_t^c} \mathbb{1}_{\hat{\boldsymbol{y}}_t^i=c} \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n'}_t^c} \mathbb{1}_{\hat{\boldsymbol{y}}_t^i=c}}, c \in \mathcal{C}_s \\
\hat{\boldsymbol{y}}_t &= \arg\min_c \operatorname{Dis}(\phi_k(\phi(\boldsymbol{x}_t), \boldsymbol{v}_t)), \\
\boldsymbol{v}_t &= \{\boldsymbol{v}_t^c\}_{c \in \mathcal{C}_s},
\end{aligned} \tag{6.32}$$

$\hat{n'}_t^c$ is the number of samples in the cth class predicted by clustering.

Entropy loss and balance coefficient in equation (6.15) is:

$$\begin{aligned}
L &= -\frac{1}{n_t} \sum_{i=1}^{n_t} B_{\hat{\boldsymbol{y}}_t^i} \hat{\boldsymbol{y}}_t^i \log(P_g(\Phi(\boldsymbol{x}_t^i))), \\
B_{\hat{\boldsymbol{y}}_t^i} &= \frac{1-\xi}{1-\xi^{\hat{n'}_t^c}}, c \in \mathcal{C}_s.
\end{aligned} \tag{6.33}$$

The whole algorithm is summarized in Algorithms 10

Algorithm 10 Multi-source-free domain adaptation under open-set.

```
Input: Target domain $\mathcal{D}_t$.
Pre-trained: Specific source model and general model.
for $\epsilon = 1, \epsilon < \mathcal{I}_t, \epsilon++,$ do
    Calculate threshold $a_o$ as in (6.29);
    Divide unknown target samples if its maximum probability value is lower than $a_o$;
    Calculate initial cluster centers and pseudo labels as in (6.30), (6.31);
    Update cluster centers and pseudo target labels as in (6.32);
    Compute balance coefficient of target domain as in (6.33);
    Compute information maximization loss as in (6.16);
    Re-train source model by freezing classifier layers as in (6.17).
end for
Predict target label as in (6.18).
Output: Target label $\boldsymbol{y}_t$.
```

6.7 Experiments

6.7.1 Datasets and Parameter Setting

This chapter employs three imbalanced real-world visual datasets to validate the proposed generally auxiliary model training method on image classification task.

Office-31 contains 4110 images collected from 31 categories, across three domains: Amazon (A), Webcam (W) and DSLR (D). Amazon has 2817 images, Webcam has 795 images, and DSLR has 498 images taken by different devices.

Office-Home collects 15588 images from 65 categories. It has four domains Art (A), Clipart (C), Product (P) and Real World (R) which share . Art has 2427 images, Clipart contains 4365 images, Product comprises 4439 images, and Real World holds 4357 images.

Office-Caltech10 consists of 2533 images from 10 categories shared by datasets Office-31 and Caltech-256. It has four domains Caltech (C), Amazon (A), Webcam (W) and DSLR (D), where Caltech has 1123 images, Amazon has 958 images, Webcam has 295 images, and DSLR has 157 images.

This work employs $ResNet50$ as the backbone ϕ, specific source feature extraction layer ϕ_k reduces the dimension of $ResNet50$ outputs from 2048 to 256. Learning rate η is $\eta = \frac{\eta_0}{(1+10\epsilon)^{0.75}}$, where $\eta_0 = 0.01$, ϵ is the training progress changing linearly from 0 to 1, the momentum is 0.9 and weight decay is $5e-4$. The smoothing parameter $\mu = 0.1$, balance parameter $\xi = 0.9999$.

6.7.2 Comparison and Analysis under Closed Set

We evaluate the proposed method on real-world visual datasets Office-31 and Office-Home. Both datasets are class imbalanced. Tasks include $A, W \to D$; $A, D \to W$; $D, W \to A$ from Office-31 and $A, C, P \to R$; $A, C, R \to P$; $A, P, R \to C$, $C, P, R \to A$ from Office-Home.

The comparison closed set methods include domain adaptation methods with and without source data. Source data available methods include:

- DAN: Deep adaptation network (Long et al., 2015);
- MRAN: Multi-representation adaptation network (Zhu et al., 2019b);
- MDDA: Manifold dynamic distribution adaptation (Wang et al., 2020b);
- DDAN: Dynamic distribution adaptation network (Wang et al., 2020b);
- ALDA: Adversarial-learned loss for domain adaptation (Chen et al., 2020);
- MFSAN: Moment matching for multi-source domain adaptation (M3SDA) (Zhu et al., 2019a);
- MSCLDA: Multi-source contribution learning for domain adaptation (Li et al., 2021c);
- LtC-MSDA: Learning to combine: knowledge aggregation for multi-source domain adaptation (Wang et al., 2020a);
- DCA: Dynamic classifier alignment for unsupervised multi-source domain adaptation (Li et al., 2022a).

Source free methods include:

- BAIT: Domain adaptation without source data by casting a bait (Yang et al., 2021b);
- PrDA: Progressive domain adaptation (Kim et al., 2020);
- SHOT: Source hypothesis transfer with information maximization (Liang et al., 2020);

- SDDA: Source data free domain adaptation- domain impression (Kurmi et al., 2021b);

- G-SFDA: Generalized source-free domain adaptation (Yang et al., 2021c).

All experiments are repeated for three times and the results are averaged accuracy. Tables 6.2 and 6.3 show the results of the proposed method and the baselines. Standard "Source data" means domain adaptation with the access of source data, while "Source free" indicates domain adaptation without source data. The results specify that the method we propose achieves the highest average classification accuracy. Results of baselines indicate that domain adaptation methods with source data generally outperform those methods without it. Except for the proposed method, few source-free methods (G-SFDA) achieve a higher average performance than the methods with source data.

6.7.3 Influence of General Model under Closed Set

Tables 6.4 and 6.5 show the results of the proposed specific source models and the global model when performing directly on the target domain directly. Tables 6.6 and 6.7 show the performance of specific and general models after fitting the source models to the target. Standard "Bef" indicates the results of performing the source models on the target domain without fitting, "Aft" designates the results of fitted source models.

The performance of the general model is higher than that of specific source models on most tasks, highlighting that the general model has better generality than specific source models. The general model is trained by treating the tasks from other sources as auxiliary tasks. Only parameters are shared to provide rich learning information, thereby improving the cross-domain ability and generality of the general model. The general model is expected to predict target labels with high

Table 6.2 : Classification accuracy (%) of the proposed generally auxiliary model and the compared methods on Office-31.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Source data	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	MRAN	99.8	96.9	70.9	89.2
	MDDA	99.2	97.1	73.2	89.8
	DDAN	**100.0**	96.7	65.3	87.3
	ALDA	**100.0**	97.7	72.5	90.1
	MFSAN	99.5	98.5	72.7	90.2
	MSCLDA	99.8	98.8	73.7	90.8
	DCA	99.6	98.9	75.1	**91.2**
Source free	ResNet	97.5	95.4	60.2	84.4
	BAIT	98.8	98.5	71.1	89.5
	PrDA	96.7	93.8	73.2	87.9
	SHOT	99.9	98.5	74.1	90.8
	SDDA	99.8	**99.0**	67.7	88.8
	GAM	99.5	98.7	**75.4**	**91.2**

Table 6.3 : Classification accuracy (%) of the proposed generally auxiliary model and the compared methods on Office-Home.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Source data	ResNet	67.8	71.3	51.8	53.4	61.1
	DAN	75.9	80.3	56.5	68.2	70.2
	MRAN	77.5	82.2	60.0	70.4	72.5
	MDDA	77.8	81.8	57.6	67.9	71.3
	DDAN	72.7	78.9	56.6	65.1	68.3
	ALDA	77.1	82.1	56.3	70.2	71.4
	MFSAN	80.8	79.0	60.7	70.0	72.6
	MetaMDA	**83.4**	81.2	60.5	70.2	73.8
	MSCLDA	80.6	79.9	61.4	71.6	73.4
	LtC-MSDA	80.1	79.2	**64.1**	67.4	72.7
	DCA	81.4	80.5	63.6	72.1	74.4
Source free	ResNet	76.3	78.8	50.1	50.9	64.0
	BAIT	77.2	79.4	59.6	71.1	71.8
	PrDA	76.8	79.1	57.5	69.3	70.7
	SHOT	81.5	83.0	57.2	72.1	73.5
	G-SFDA	82.2	**83.4**	57.9	72.0	73.9
	GAM	83.1	83.1	60.1	**73.5**	**75.0**

quality at the beginning of fitting the source model, which is helpful to learn cluster centers to obtain more correct pseudo labels.

Table 6.4 : Source-only classification accuracy (%) of specific and general models on Office-31.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Bef	S1	96.6	95.9	63.2	85.2
	S2	98.4	96.1	63.5	86.0
	G	**98.6**	**96.4**	**64.9**	**86.6**

Table 6.5 : Source-only classification accuracy (%) of specific and general models on Office-Home.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Bef	S1	80.0	75.0	52.6	63.5	67.8
	S2	78.5	76.0	52.0	63.9	67.6
	S3	77.1	**79.1**	51.5	65.2	68.2
	G	**80.7**	78.0	**53.8**	**65.8**	**69.6**

With the process of re-training, as shown in Tables 6.6 and 6.7, specific source models and general model perform similarly on the target domain. Since the classifiers are frozen in re-training, when the training is convergent, the performance of all specific source models become stable. As a linear combination of specific source models, the performance of the general model will also be stable, and it is close to the prediction of local models.

Table 6.6 : Classification accuracy (%) of specific and general models on Office-31 after fitting.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Aft	S1	99.3	**98.7**	**75.4**	91.1
	S2	**99.5**	98.2	75.3	91.0
	G	**99.5**	**98.7**	**75.4**	**91.2**

Table 6.7 : Classification accuracy (%) of specific and general models on Office-Home after fitting.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Aft	S1	**83.1**	83.1	60.0	73.2	74.9
	S2	82.8	82.8	**60.1**	73.4	74.8
	S3	82.9	**83.2**	**60.1**	**73.7**	**75.0**
	G	**83.1**	83.1	**60.1**	73.5	**75.0**

6.7.4 Influence of Class Balanced Coefficient under Closed Set

Tables 6.8, 6.9 and 6.10 show the performance of the general and source local models with and without class balanced coefficient before and after fitting the model to the target domain. Standard "Bef" indicates the accuracy before fitting the source model, while "Aft" indicates that after fitting. "N" is the model trained without class balanced coefficient, and "Y" is that trained with balanced coefficient. It can be seen that the model trained with class balanced coefficient achieves higher performance than that without it, meaning that balancing the class data has a positive influence on the learning of both specific and general models.

Table 6.8 : Classification accuracy (%) of the general model with and without class balanced coefficient. "Bef" means source only model, "Aft" is adapted model.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Bef	N	99.2	95.9	64.2	86.4
	Y	98.6	96.4	64.9	86.6
Aft	N	**99.8**	98.6	73.3	90.6
	Y	99.5	**98.7**	**75.4**	**91.2**

6.7.5 Influence of Distance Measurements in Deep Clustering

We employ cosine similarity to measure the distance between a target sample and the target clusters when predicting pseudo labels (in equation (6.12) and (6.13)). Cosine distance has the advantage of defining the similarity between two multi-dimensional features by returning the angle of data, while the Euclidean distance can be far. To validate the superiority of cosine distance in this situation, Table 6.11 shows the results of the proposed method using cosine distance and Euclidean dis-

Table 6.9 : Classification accuracy (%) of the specific source model 1 with and without class balanced coefficient. “Bef” means source only model, “Aft” is adapted model.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Bef	N	98.4	86.5	62.9	82.6
	Y	96.6	95.9	63.2	85.2
Aft	N	**99.8**	97.5	71.5	89.6
	Y	99.3	**98.7**	**75.4**	**91.1**

Table 6.10 : Classification accuracy (%) of the specific source model 2 with and without class balanced coefficient. “Bef” means source only model, “Aft” is adapted model.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Bef	N	99.2	96.9	64.6	86.9
	Y	98.4	96.1	63.5	86.0
Aft	N	**99.8**	**98.7**	73.2	90.6
	Y	99.5	98.2	**75.3**	**91.0**

tance, it shows that the model based on cosine distance achieves higher classification accuracy than that with Euclidean distance.

Table 6.11 : Classification accuracy (%) of specif and general models on Office-Home with different distance measurements.

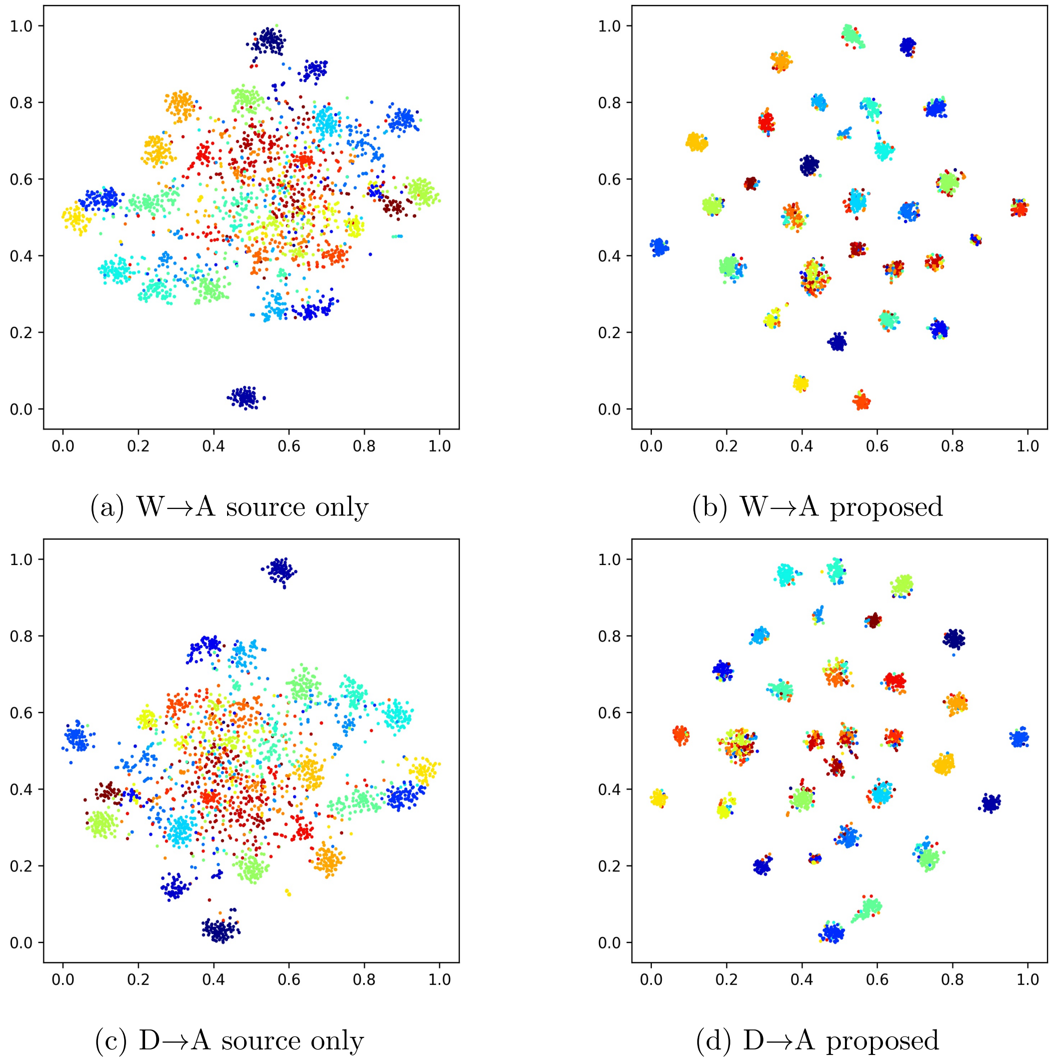
Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Euclidean	S1	82.8	82.1	59.9	72.5	74.3
	S2	82.5	81.6	**60.5**	72.6	74.3
	S3	82.5	82.3	60.3	72.8	74.5
	G	**83.1**	82.0	60.3	72.6	74.5
Cosine	S1	**83.1**	83.1	60.0	73.2	74.9
	S2	82.8	82.8	60.1	73.4	74.8
	S3	82.9	**83.2**	60.1	**73.7**	**75.0**
	G	**83.1**	83.1	60.1	73.5	**75.0**

6.7.6 Data Visualization under Closed Set

Figures 6.2 and 6.3 show the T-SNE visualization (Maaten and Hinton, 2008) of the proposed method. Different colors indicates the classes. Taking tasks $W, D \rightarrow A$ and $C, P, R \rightarrow A$ as examples, compared with performing a source model on the target domain directly, the proposed method can divide most target samples correctly with clear boundaries.

6.7.7 Comparison and Analysis under Closed Set with Sample and Source Distillation

To explore how the quality of source samples and domains effects the performance of the proposed generally auxiliary model method under source-free setting, we



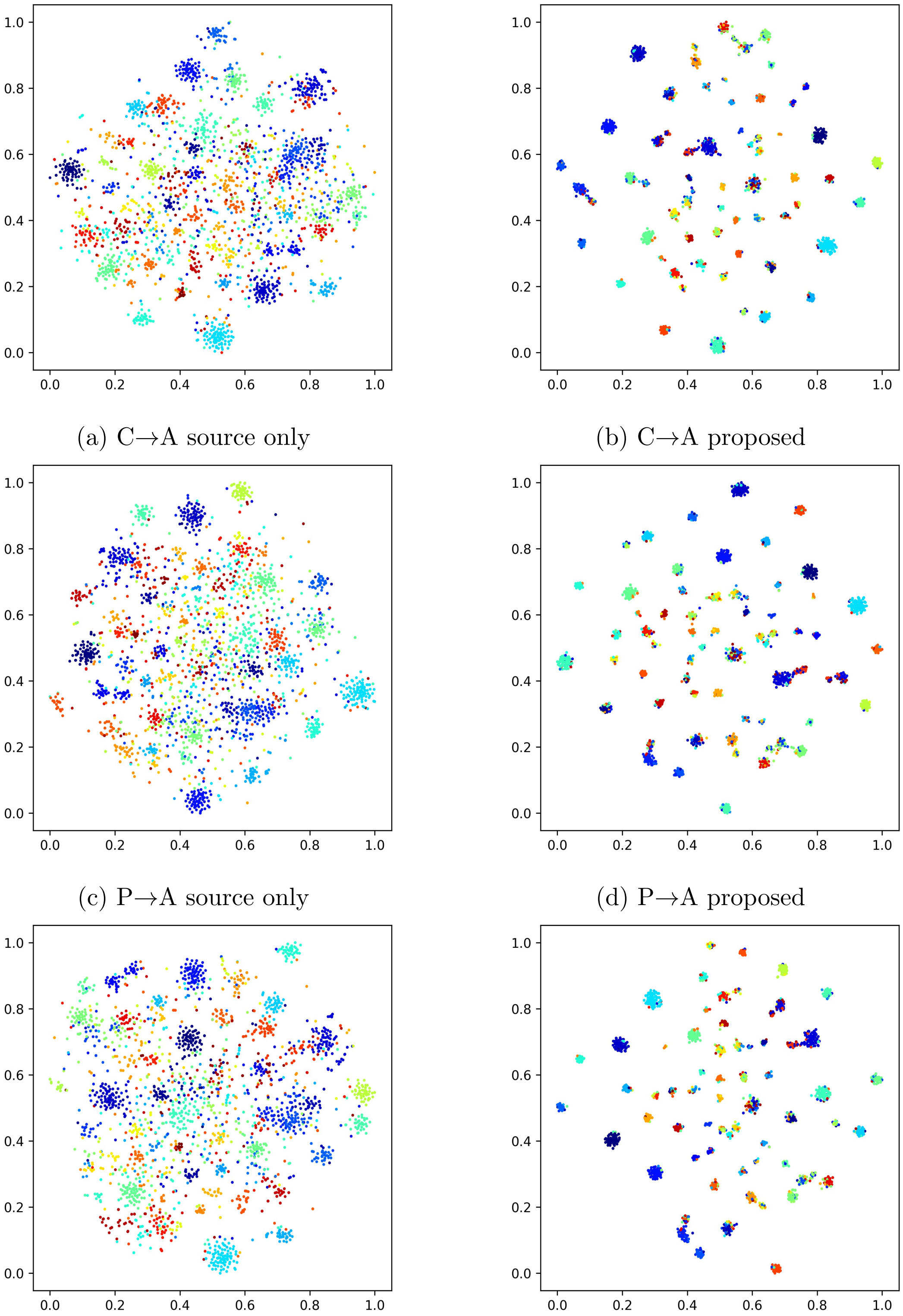
(a) W→A source only

(b) W→A proposed

(c) D→A source only

(d) D→A proposed

Figure 6.2 : T-SNE visualization of Office-31 under closed set.



(a) C→A source only

(b) C→A proposed

(c) P→A source only

(d) P→A proposed

(e) R→A source only

(f) R→A proposed

Figure 6.3 : T-SNE visualization of Office-Home under closed set.

distill inefficient samples and sources from datasets Office-31 and Office-Caltech10 to validate the proposed method. Transfer tasks include $A, W \rightarrow D$; $A, D \rightarrow W$; $D, W \rightarrow A$ from Office-31, and $A, D, W \rightarrow C$; $C, D, W \rightarrow A$; $A, C, D \rightarrow W$, $A, C, W \rightarrow D$ from Office-Caltech10.

The comparison closed set methods include domain adaptation methods with and without source data. Source data available methods include:

- DAN: Deep adaptation network (Long et al., 2015);
- MRAN: Multi-representation adaptation network (Zhu et al., 2019b);
- MDDA: Manifold dynamic distribution adaptation (Wang et al., 2020b);
- DDAN: Dynamic distribution adaptation network (Wang et al., 2020b);
- ALDA: Adversarial-learned loss for domain adaptation (Chen et al., 2020);
- MFSAN: Moment matching for multi-source domain adaptation (M3SDA) (Zhu et al., 2019a);
- MSCLDA: Multi-source contribution learning for domain adaptation (Li et al., 2021c);
- LtC-MSDA: Learning to combine: knowledge aggregation for multi-source domain adaptation (Wang et al., 2020a);
- DCA: Dynamic classifier alignment for unsupervised multi-source domain adaptation (Li et al., 2022a).

Source free methods include:

- BAIT: Domain adaptation without source data by casting a bait (Yang et al., 2021b);

- PrDA: Progressive domain adaptation (Kim et al., 2020);
- SHOT: Source hypothesis transfer with information maximization (Liang et al., 2020);
- SDDA: Source data free domain adaptation- domain impression (Kurmi et al., 2021b);
- CDCL: Cross-domain contrastive learning for unsupervised domain adaptation (Wang et al., 2022);
- DECISION: Unsupervised multi-source domain adaptation without access to source data (Ahmed et al., 2021).

Experiment results are shown in Table 6.12 and 6.13. It can be seen the proposed method achieves higher performance than most compared methods. On dataset Office31, the proposed method gains the highest accuracy on all tasks, while on data Office-Caltech10, it achieves the highest average accuracy. It also indicates that the proposed method under a source-free setting gains higher performance than that with source data. It is because the model trained with source data is dominated by source domains, which means only invariant information is extracted. However, in source-free domain adaptation, both invariant and specific information of the target domain are employed. In addition, the source-free method employs more target pseudo labels than the proposed method with source data, which uses at most half of the target pseudo labels.

Table 6.14 shows the influence of sample quality. “Source” means the source domain with all samples, “Distilled Source” means source domain with selected samples. “S1” and “S2” indicate specific source models, “G” indicates generally auxiliary model. It can be seen that with sample and source distillation, both specific and general source models achieve the higher performance than the models

Table 6.12 : Accuracy (%) on dataset Office31 of the proposed and comparison methods under source-free domain adaptation.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Source data	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	MRAN	99.8	96.9	70.9	89.2
	MDDA	99.2	97.1	73.2	89.8
	DDAN	**100.0**	96.7	65.3	87.3
	ALDA	**100.0**	97.7	72.5	90.1
	MFSAN	99.5	98.5	72.7	90.2
	MSCLDA	99.8	98.8	73.7	90.8
	DCA	99.6	98.9	75.1	91.2
Source free	ResNet	97.5	95.4	60.2	84.4
	BAIT	98.8	98.5	71.1	89.5
	PrDA	96.7	93.8	73.2	87.9
	SHOT	94.9	97.8	75.0	89.2
	SDDA	99.8	99.0	67.7	88.8
	CDCL	97.2	95.3	75.3	89.3
	DECISION	99.6	98.4	75.4	91.1
	GAM+SSD	**100.0**	**99.6**	**75.8**	**91.8**

Table 6.13 : Accuracy (%) on dataset Office-Caltech10 of the proposed and comparison methods under source-free domain adaptation.

Standards	Method	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
Source data	ResNet	82.5	91.2	98.9	99.2	93.0
	MFSAN	93.8	95.1	99.1	98.7	96.7
	MSCLDA	94.1	95.3	99.1	98.5	96.8
	DCA	94.7	96.0	99.7	99.1	97.4
Source free	ResNet	92.1	96.3	98.0	99.5	96.5
	BAIT	95.7	**97.5**	98.0	97.5	97.2
	PrDA	94.6	97.3	97.6	97.1	96.7
	SHOT	95.8	95.7	99.6	96.8	97.0
	DECISION	**95.9**	95.9	99.6	**100.0**	**98.0**
	GAM+SSD	95.8	96.0	**100.0**	**100.0**	**98.0**

without sample and source selection.

Table 6.14 : Accuracy (%) on dataset Office31 of the source only model with and with out distillation.

Standards	Method	A, W→D	A, D→W	W, D→A	Avg
Source	S1	98.6	95.7	65.0	86.4
	S2	98.4	95.0	65.2	86.2
	G	98.6	95.5	66.0	86.7
Distilled Source	S1	99.0	96.6	65.9	87.2
	S2	99.0	**96.7**	66.3	87.3
	G	**99.2**	**96.7**	**66.7**	**87.5**

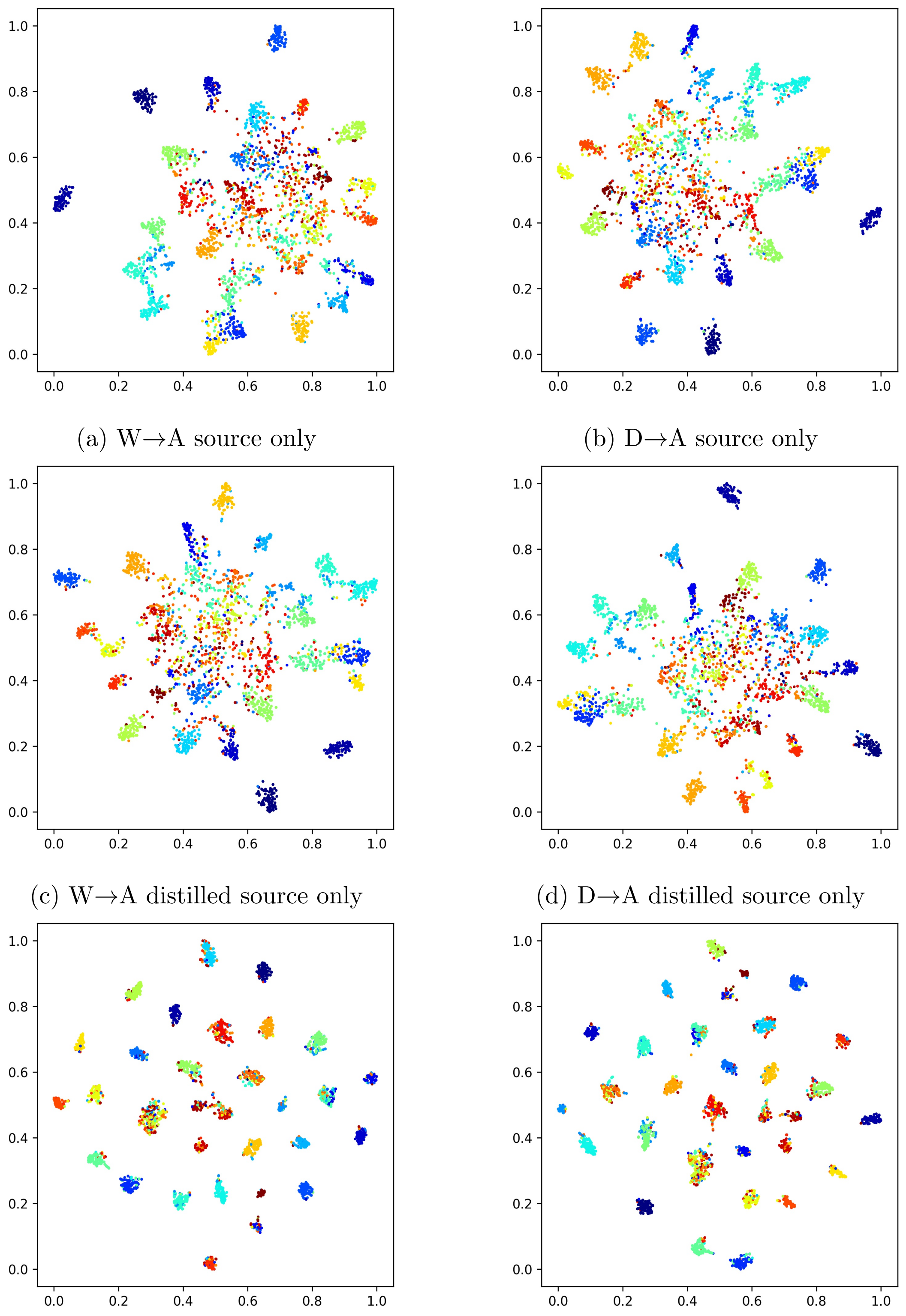
6.7.8 Data Visualization under Closed Set with Sample and Source Distillation

Figure 6.4 shows the data visualization of the proposed method with sample and source selection. Compared Figures 6.4(a) and 6.4(b) with 6.4(c) and 6.4(d), it can been seen that distilling source samples dose not degrade the classification performance. Figures 6.4(e) and 6.4(f) indicate the proposed method after adaptation based on distilled source domains divides most samples correctly.

6.7.9 Comparison and Analysis under Partial Set

We validate the proposed method on dataset Office-Home, transfer tasks include $A, C, P \rightarrow R$; $A, C, R \rightarrow P$; $A, P, R \rightarrow C$, $C, P, R \rightarrow A$, where source domains contain 65 categories, while target domain contains 25 categories.

The partial domain adaptation methods used for comparison include:



(a) W→A source only

(b) D→A source only

(c) W→A distilled source only

(d) D→A distilled source only

(e) W→A proposed

(f) D→A proposed

Figure 6.4 : T-SNE visualization with sample and source distillation.

- SAN: Partial transfer learning with selective adversarial networks (Cao et al., 2018a);
- ETN: Learning to transfer examples for partial domain adaptation (Cao et al., 2019);
- SAFN: Adaptive feature norm approach for unsupervised domain adaptation (Xu et al., 2019b);
- DARL: Domain adversarial reinforcement learning for partial domain adaptation (Chen et al., 2022b)
- MSAN: Attention guided for partial domain adaptation (Zhang and Zhao, 2021);
- SHOT: Source hypothesis transfer with information maximization (Liang et al., 2020).

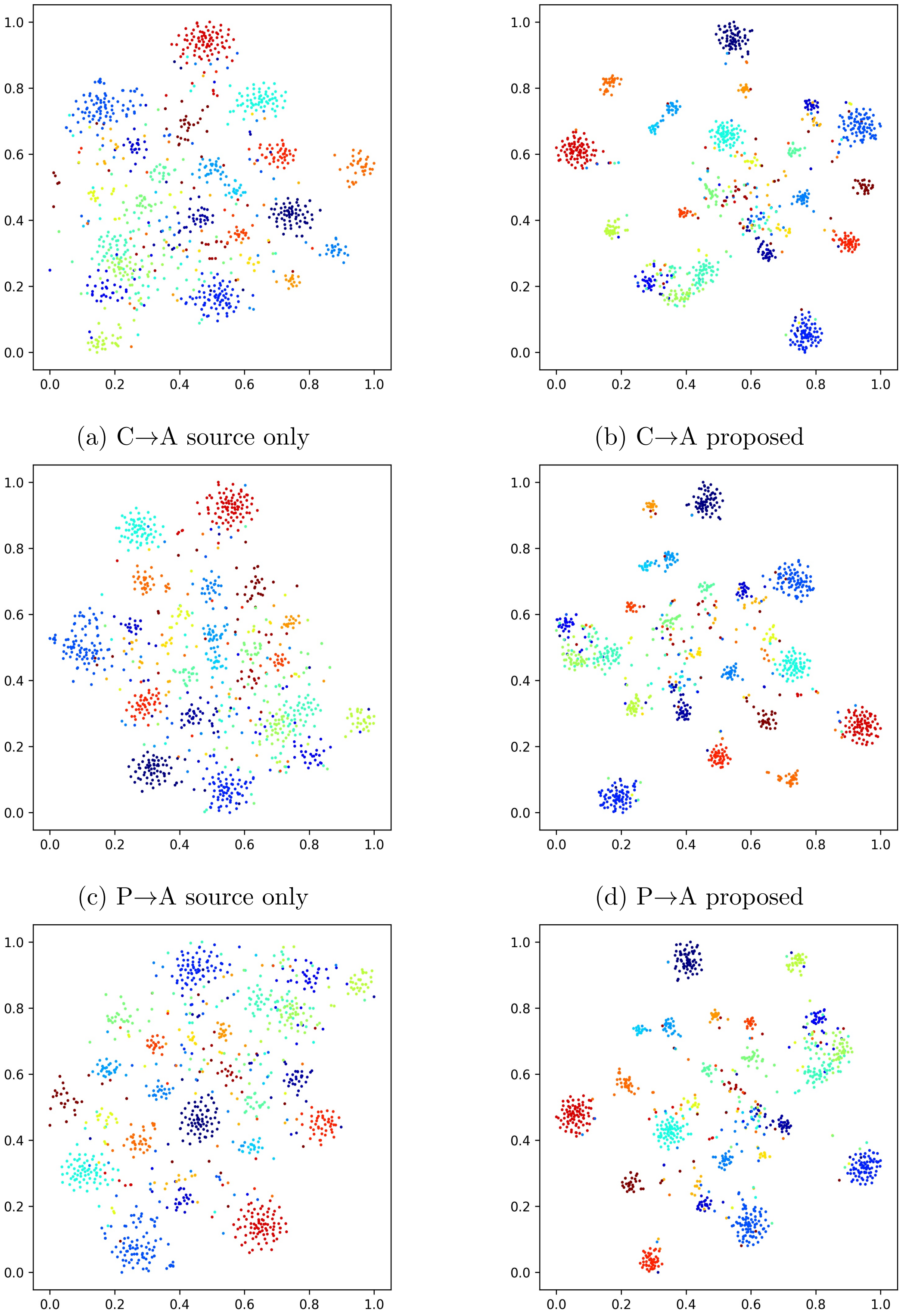
Table 6.15 shows the results of applying the proposed method to partial domain adaptation, respectively. The proposed method achieves highest performance on most tasks under both settings, indicating the superiority of multi-source domains and the generally auxiliary model.

6.7.10 Data Visualization under Partial Set

Figure 6.5 shows the data visualization of the proposed method under partial set, where source label space is larger than target label space.

6.7.11 Comparison and Analysis under Open-Set

We validate the proposed method on dataset Office-Home, transfer tasks include $A, C, P \to R$; $A, C, R \to P$; $A, P, R \to C$, $C, P, R \to A$, where source domains contain 65 categories, while target domain contains 25 categories.



(a) C→A source only

(b) C→A proposed

(c) P→A source only

(d) P→A proposed

(e) R→A source only

(f) R→A proposed

Figure 6.5 : T-SNE visualization under partial domain adaptation.

Table 6.15 : Comparison of classification accuracy (%) on Office-Home under partial domain adaptation.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Source data	ResNet	71.2	67.2	45.4	61.6	61.4
	SAN	77.5	70.8	46.4	66.5	65.3
	ETN	79.6	75.7	57.4	69.0	70.4
	SAFN	79.9	76.4	58.2	72.9	71.9
	DARL	84.2	77.5	54.5	72.0	72.1
	MSAN	80.4	76.2	56.7	67.2	70.1
Source free	SHOT	88.4	82.4	64.0	77.6	78.1
	GAM	**91.1**	**85.9**	**68.7**	**81.5**	**81.8**

Comparison open-set domain adaptation methods include:

- OSBP: Open set domain adaptation by backpropagation (Saito et al., 2018);
- STA: Separate to adapt: Open set domain adaptation via progressive separation (Liu et al., 2019a);
- DAOD: Open set domain adaptation: theoretical bound and algorithm (Fang et al., 2021);
- SHOT: Source hypothesis transfer with information maximization (Liang et al., 2020);
- PGL: Source-free progressive graph learning for open-set domain adaptation (Luo et al., 2022).

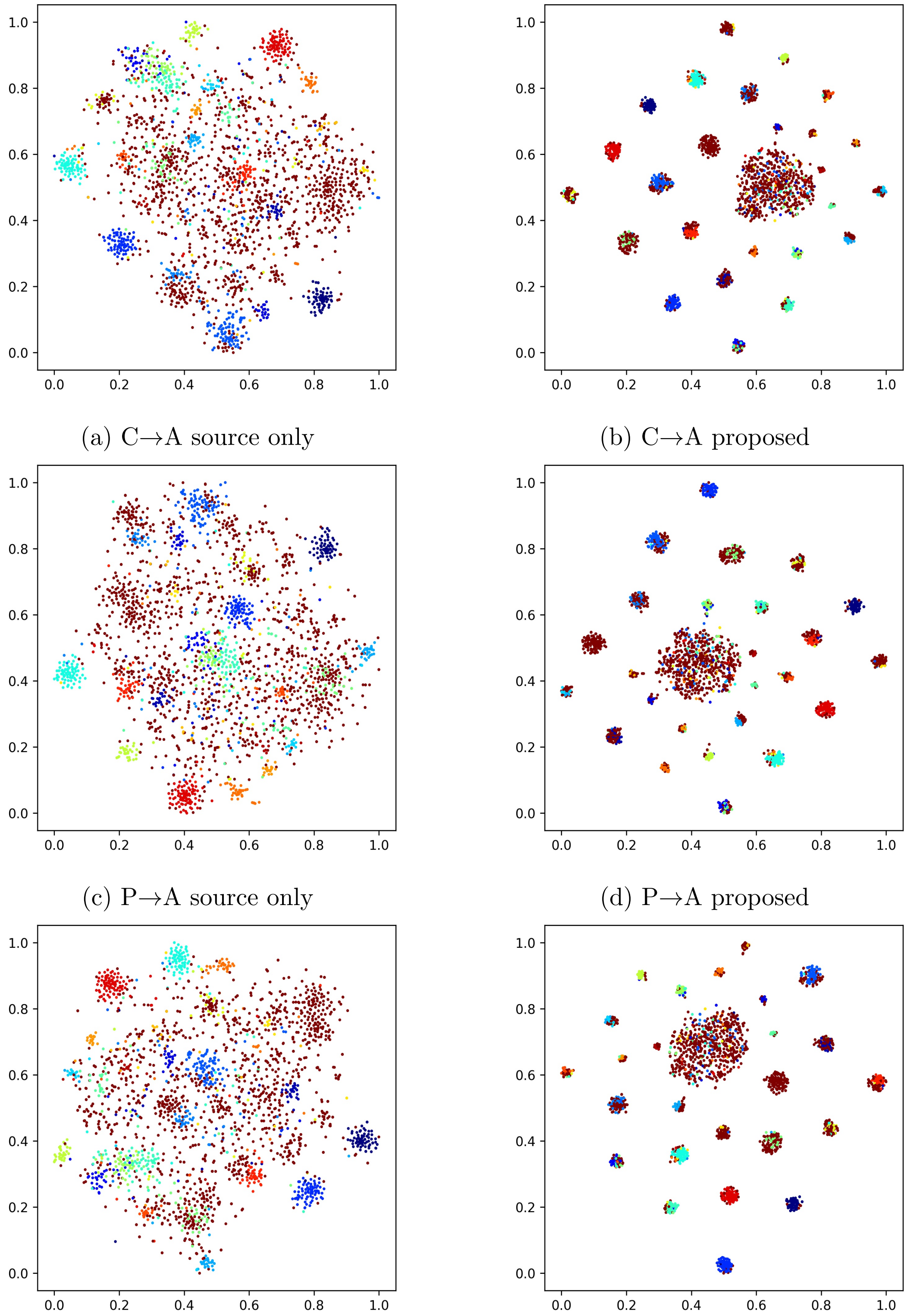
Table 6.16 shows the results of applying the proposed method to partial and open-set domain adaptation, respectively. The proposed method achieves highest performance on most tasks under both settings, indicating the superiority of multi-source domains and the generally auxiliary model.

Table 6.16 : Comparison of classification accuracy (%) on Office-Home under open-set domain adaptation.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Source data	ResNet	65.3	63.1	62.8	69.8	65.3
	OSBP	64.2	62.8	65.4	70.2	65.7
	STA	70.4	67.7	66.4	**73.3**	69.5
	DAOD	79.8	73.5	58.6	67.2	69.8
Source free	SHOT	82.4	79.3	61.4	64.5	71.9
	PGL	**86.1**	79.2	63.8	**75.1**	76.1
	GAM	84.7	**83.0**	**66.5**	73.3	**76.9**

6.7.12 Data Visualization under Open-Set

Figure 6.6 shows the data visualization of the proposed method under open-set domain adaptation, where source label space is a subset of target label space. Taking task $C, P, R \rightarrow A$ as an example, it can be seen the proposed method can identify both known and unknown classes.



(a) C→A source only

(b) C→A proposed

(c) P→A source only

(d) P→A proposed

(e) R→A source only

(f) R→A proposed

Figure 6.6 : T-SNE visualization under open-set domain adaptation.

6.8 Summary

In this chapter, we propose a multi-source-free domain adaptation method with generally auxiliary model training. It constructs a global model from multiple source domains as an auxiliary task to improve the cross-domain ability and generality of the source models. In addition, the class balanced coefficient is introduced to ensure the classification performance on the classes which containing fewer samples. Experiments on real-world datasets on both homogeneous and heterogeneous label spaces with and without sample and source distill show the superiority of the proposed method. Under closed set, on dataset Office-31, the accuracy of the proposed method is improved by 0.4% compared with the baselines, and on dataset Office-Home, the accuracy of the proposed method increases by 1.1%.

Combining specific source models has a positive influence on the transferring in most situations. But multiple source domains requires more parameters and computer memory to train the models. Experiments also show that the global model may introduce negative transfer on some tasks. We will take these problems as a focus of future study to explore a more efficient method to transfer information from multiple source domains to the target domain.

Chapter 7

Source-Free Multi-Domain Adaptation with Fuzzy Rule-Based Deep Neural Networks

7.1 Introduction

Neither existing domain adaptation methods with source data nor the methods without source data ignore the soft information caused by uncertain data during transfer. To solve this problem, fuzzy domain adaptation attracts attention in light of its advantages in building soft information to handle data uncertainty (Lu et al., 2015). A theory-based study- learning from imprecise observations- investigates multi-class classification with fuzzy observations and protects data privacy by transferring original data into concepts, thereby considering both data uncertainty and security (Ma et al., 2021a). It creates fuzzy vectors from real observations and provides an estimation error bound learned from fuzzy random variables. Fuzzy multi-source transfer learning focuses on selecting and merging fuzzy rules from multiple domains to generate target rules under both homogeneous and heterogeneous domain adaptation scenarios (Lu et al., 2020). Interactive transfer learning distills useless source information by a knowledge filter, and designs a self-balancing mechanism to learn the scene difference and inherent uncertainty, which are used match source and target domains by reducing unbalanced diversity (Han et al., 2021).

Previous fuzzy domain adaptation methods focus on transferring the invariant knowledge extracted from data, but fuzzy domain adaptation without source data remains unsolved. Most source-free domain adaptation methods rarely take the inherent soft information into account, especially in deep neural networks, where the data is trained over the batch by extracting region information using convolution kernels. The extracted regions and samples belonging to the same category but from different scales contain multiple information levels, while samples from different categories can contain similar regions, which contribute differently to its category and the whole classifier. In this situation, dividing samples into multiple groups according to their information levels might benefit the classification. The fuzzy model has the advantage of describing the degrees of multi-level information belonging to multiple categories. Hence, in this chapter, we propose source-free multi-domain adaptation with fuzzy rule-based deep neural networks (SF-FDN) to extract soft information from precise data, which introduces fuzzy C-means clustering and Takagi-Sugeno fuzzy rules to source-free domain adaptation. The proposed method improves the generality of a source private model on multiple domains by establishing auxiliary tasks, which derive benefit from similar tasks and preserve the data privacy in source domains simultaneously. When transferring source rules and parameters to a target domain, to guarantee the accuracy of pseudo labels, a target sample selection strategy is adopted to collect pseudo labels with low noise. We use the pseudo labels to supervise the training of the target model on the label-level, and develop anchor-based alignment to reduce data bias between domains on the distribution-level. This approach allows us to extract both invariant and specific information of target domains to parameterize the target model. Our contributions are summarized as follows:

- We propose source-free multi-domain adaptation with fuzzy rule-based deep neural networks. To the best of our knowledge, this is the first work adopting fuzzy rules to deal with source-free transfer learning. The proposed method deals with soft information to enrich transferable knowledge among both classes and domains, which most non-fuzzy methods rarely consider. It develops fuzzy

C-means clustering in a deep structure to construct fuzzy rules and introduces the Takagi-Sugeno model to solve domain adaptation without source data. Based on experiments, the fuzzy source-free domain adaptation method is superior to non-fuzzy methods by transferring knowledge on multiple information levels;

- We develop an auxiliary learning mechanism to enhance the multi-domain performance of the private source models. Few existing source-free methods handle multiple source domains. The proposed method takes advantage of category information from other source domains by jointly training source parameters. Compared with existing source-free multi-domain adaptation methods which train source models independently, by doing this, multiple source domains can now share invariant knowledge without sharing data;

- We generate source anchors from source fuzzy rules to collect highly representative class features, which are employed to define an anchor-based alignment strategy to fit the pre-trained source model to the target domain while protecting the source data. By reducing the distance between source and target anchors which highly represent class information, the target feature extractor is forced to transform target data into the latent feature space which is closer to source distribution. Compared with existing source data generation methods, the proposed source anchors based on fuzzy rules can collect more usable knowledge on multiple information levels;

- We build a selection strategy in assistance with fuzzy outputs and nearest clustering to collect strong target samples to calculate clustering centers which we employ to predict pseudo labels with high confidence. In comparison to existing methods computing clustering centers using all pseudo target labels, the proposed strategy can reduce the label noise which is known to result in negative transfer.

7.2 Problem Setting and Notations

In this chapter, we focus on data-free domain adaptation with multiple source domains for the image classification task. Table 7.1 describes the symbols in this chapter.

Table 7.1 : Symbol descriptions.

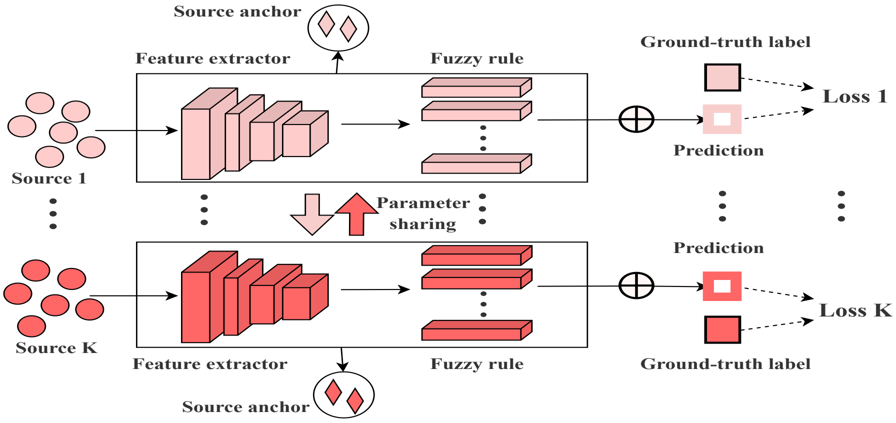
Notation	Description
$\mathcal{D}_{s_k}$, $\mathcal{D}_t$	the private source/target domain
n_{s_k}, n_t	number of source/target samples
$\boldsymbol{x}_{s_k}$, $\boldsymbol{x}_t$	source/target sample
$\boldsymbol{y}_{s_k}$	source label of $\boldsymbol{x}_{s_k}$
ϕ	pre-trained deep-structured backbone
ϕ_k	the kth source private feature extractor
$\boldsymbol{v}^l_{s_k}$, $\boldsymbol{v}^l_{t_k}$	clustering prototype from the source/target domain
$u^l_{s_k}$, $u^l_{t_k}$	membership of the source/target sample belonging to lth fuzzy set
$\boldsymbol{f}^c_{s_k}$, $\boldsymbol{f}^c_{t_k}$	class anchor from the source/target domain
$\boldsymbol{w}_c$	deep clustering center from target domain
$\boldsymbol{r}$	probability vector estimated by classifier

7.3 The Proposed Fuzzy Source-Free Multi-Domain Adaptation Method

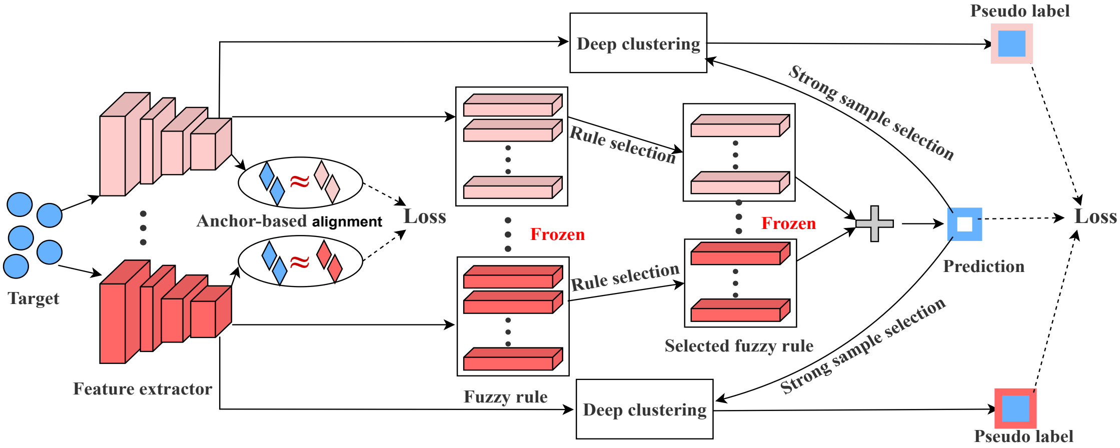
We design a new fuzzy rule-based deep neural network to tackle data-free domain adaptation with multiple source domains. The proposed method is illustrated in Fig. 7.1. The top figure displays the process of source private model training with auxiliary tasks, while the bottom figure indicates the model adaptation based on self-supervised training. As shown in Fig. 7.1(a), for each source domain, the original data is first transformed into a latent feature space by the feature extractor. To leverage soft information, we adopt fuzzy C-means clustering to calculate prototypes of each source domain and memberships of samples to define fuzzy rules and predict the final outputs. Source anchors are generated based on the fuzzy model to describe class information distinctly and preserve data privacy for other users. The error between fuzzy outputs and the ground-truth labels are employed to parameterize the training. To provide better generality of source private models in predicting target task, source parameters and fuzzy rules are shared among domains as auxiliary models for each other using a joint training method. In Fig. 7.1(b), given the pre-trained source models based on fuzzy rules, pre-learned source rules are frozen to match target data to source distribution where source models can be transferred, while feature extractors are re-trained. Anchor-based alignment is designed to force the target data to source feature space by extracting invariant information. Besides, to extract specific information from target domain, self-supervision is constructed to parameterize the re-training. The success of self-supervision relies on the high quality of pseudo labels. To guarantee these pseudo labels provided by deep clustering come with low noise, a sample selection strategy based on fuzzy outputs is built to select target labels confidently and generate reliable clustering centers. The cross-entropy loss between pseudo labels and the fuzzy outputs is employed to fine-tune the feature extractors.

7.3.1 Source Private Model Training

In this chapter, we employ Takagi-Sugeno fuzzy rules to build the source model. Given source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$, for input data $\boldsymbol{x}_{s_k} \in \mathbb{R}^s$ and the corresponding



(a) Source private model training with auxiliary learning.



(b) Model adaption with self-supervised learning.

Figure 7.1 : The procedure of the proposed method. The solid arrow means data-flow, the dashed arrow means loss computing. Figure (a) indicates source model training. Auxiliary tasks are constructed by sharing source parameters. Fuzzy C-means clustering is employed to learn the prototypes and memberships to build the fuzzy rules. Source anchors are extracted to describe the source class information without referring to the original data. Figure (b) demonstrates domain adaptation. By freezing source rules, self-supervised learning is employed to fine-tune feature extractors. Anchor-based alignment is built to match domains on the data-level by extracting invariant information. Deep clustering and a sample selection strategy are designed to predict pseudo labels with low noise which learn specific target information.

output $\boldsymbol{y}_{s_k} \in \mathbb{R}^c$ in the kth source domain, a rule can be described as:

$$\begin{aligned}&\text{if } \boldsymbol{x}_{s_k} \textit{ is } A_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))),\\&\text{then } \boldsymbol{y}_{s_k} \textit{ is } P_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))),\\&l = 1, 2, \cdots, L_k.\end{aligned}$$

ϕ_k is the private feature extractor in the kth source domain, while ϕ is a pre-trained deep-structured backbone on a very large dataset. Parameters of ϕ are shared among all domains. Feature extractors transform original data to feature space $\mathbb{R}^d$. A_{kl} represents the fuzzy condition of the lth rule, P_{kl} is a function transforming data from $\mathbb{R}^d$ to $\mathbb{R}^c$. L_k represents the number of rules in the kth source domain.

The final prediction of the Takagi-Sugeno fuzzy model in each source domain is the linear combining of the outputs of all rules, which is:

$$\boldsymbol{y}_{s_k} = \sum_{l=1}^{L_k} u_{s_k}^l \cdot P_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tag{7.1}$$

$u_{s_k}^l$ is the membership of data $\boldsymbol{x}_{s_k}$ belonging to the lth fuzzy set.

There are three problems to be solved to build the source model: first, how to define fuzzy rule number L_k; second, how to learn function P_{kl}; third, how to measure the membership $u_{s_k}^l$ to calculate the final prediction.

To solve the first problem, we design a class grouping strategy based on the similarities among each pair of classes. Here, the correlation coefficient is employed to measure the similarity between every two classes. Denote any class pair as $(\boldsymbol{x}_{c_i}, \boldsymbol{x}_{c_j})$, the correlation coefficient between each two classes is calculated as:

$$\rho_{ij} = \frac{\mathbb{E}(\boldsymbol{x}_{c_i}\boldsymbol{x}_{c_j}) - \mathbb{E}(\boldsymbol{x}_{c_i})\mathbb{E}(\boldsymbol{x}_{c_j})}{\sqrt{\mathbb{E}(\boldsymbol{x}_{c_i}^2) - (\mathbb{E}(\boldsymbol{x}_{c_i}))^2}\sqrt{\mathbb{E}(\boldsymbol{x}_{c_j}^2) - (\mathbb{E}(\boldsymbol{x}_{c_j}))^2}}, \tag{7.2}$$

where

$$\boldsymbol{x}_c = \frac{\sum_{i=1}^{n_{s_k}^c} \mathbb{1}_{\boldsymbol{y}_{s_k}^i = c} \cdot \phi(\boldsymbol{x}_{s_k}^i)}{\sum_{i=1}^{n_{s_k}^c} \mathbb{1}_{\boldsymbol{y}_{s_k}^i = c}}, \tag{7.3}$$

$n^c_{s_k}$ denotes the number of source samples in the cth class. When $\rho_{ij} > a_\rho$, where a_ρ is a threshold, we think classes c_i and c_j are similar, and they can share the same rule. Here, we use $\boldsymbol{Y}_l$ to denote the label set containing similar classes.

To solve the second problem, we apply a structural risk minimization principle (Vapnik, 1999) to learn the function. In the classification task, P_{kl} is a classifier parameterized by minimizing the assumption between the prediction and the ground-truth source labels, which can be expressed as:

$$P_{kl} = \underset{\substack{P_{s_{kl}} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}_{s_k}}}{\arg\min} \mathcal{L}(P_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \boldsymbol{y}_{s_k}), \tag{7.4}$$

where

$$\mathcal{L} = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}^i_{s_k} \log(P_{kl}(\phi_k(\phi(\boldsymbol{x}^i_{s_k}))). \tag{7.5}$$

To improve the training speed and prevent source model parameters from overfitting which may fail the transfer, a label smoothing strategy is adopted to transform hard labels to soft labels (Müller et al., 2019; Liang et al., 2020), which is:

$$\tilde{\boldsymbol{y}}_{s_k} = (1 - \mu)\boldsymbol{y}_{s_k} + \mu/C, \tag{7.6}$$

where μ is the smoothing parameter, and C is the number of source classes. Classifier P_{kl} in equation (7.4) with smooth label is:

$$P_{kl} = \underset{\substack{P_{kl} \\ (\boldsymbol{x}_{s_k}, \tilde{\boldsymbol{y}}_{s_k}) \in \mathcal{D}_{s_k}}}{\arg\min} \mathcal{L}(P_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}), \tag{7.7}$$

where $\mathcal{L}$ is re-written as:

$$\mathcal{L} = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \tilde{\boldsymbol{y}}^i_{s_k} \log(P_{kl}(\phi_k(\phi(\boldsymbol{x}^i_{s_k})))). \tag{7.8}$$

To solve the third problem, fuzzy C-mean clustering is adopted which is a popular technique for calculating the memberships. Setting the cluster number as the rule number defined using equation (7.2), it calculates a prototype in every cluster to estimate data membership. Generally, the cluster prototypes and data memberships are updated alternately by fixing the other. In this work, cluster prototypes are initialized as the mean values of samples from the same cluster grouped according to their labels, expressed as:

$$\boldsymbol{v}_{s_k}^l = \frac{\sum_{i=1}^{n_{s_k}^l} \mathbb{1}_{\boldsymbol{y}_{s_k}^i \in \boldsymbol{Y}_l} \cdot \phi_k(\phi(\boldsymbol{x}_{s_k}^i))}{\sum_{i=1}^{n_{s_k}^l} \mathbb{1}_{\boldsymbol{y}_{s_k}^i \in \boldsymbol{Y}_l}}, \tag{7.9}$$

$\boldsymbol{Y}_l$ is a label set containing similar classes. $n_{s_k}^l$ is the number of samples in the label set $\boldsymbol{Y}_l$. Given cluster prototypes $\{\boldsymbol{v}_{s_k}^l\}_{l=1}^{L_k}$, the membership of data $\boldsymbol{x}_{s_k} \in A_{kl}$ is generally defined as:

$$u_{s_k}^l = \frac{1}{\sum_{i=1}^{L_k} (\frac{\|\boldsymbol{v}_{s_k}^l - \phi_k(\phi(\boldsymbol{x}_{s_k}))\|}{\|\boldsymbol{v}_{s_k}^i - \phi_k(\phi(\boldsymbol{x}_{s_k}))\|})^{\frac{2}{m-1}}}. \tag{7.10}$$

Using the membership calculated via equation (7.10), the cluster prototypes are updated with training processing as:

$$\boldsymbol{v}_{s_k}^l = \frac{\sum_{i=1}^{n_{s_k}} (u_{s_k}^{li})^m \cdot \phi_k(\phi(\boldsymbol{x}_{s_k}^i))}{\sum_{i=1}^{n_{s_k}} (u_{s_k}^{li})^m}. \tag{7.11}$$

The loss function of training classifiers defined by fuzzy rules in each source domain is:

$$\begin{aligned}\mathcal{L}_k =& \mathcal{L}(P_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}) + \\ & \mathcal{L}(\sum_{l=1}^{L_k} u_{s_k}^l \cdot P_{kl}(\phi_k(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}).\end{aligned}$$

To enhance the performed generality across tasks of the private source models, the auxiliary learning strategy is designed to training multiple source models jointly, and is expected to make full use of the classification information from other source domains. Not all fuzzy rules from a source domain can be performed on other different source domains because of data shift. Hence, for the kth source domain, we choose half nearest rules (denote as L_{near}) from different source domains as auxiliary tasks to improve the generality of source models. The auxiliary tasks are trained as:

$$\mathcal{L}_{aux} = \frac{1}{K-1}\sum_{k' \neq k}^{K} \mathcal{L}\Big(\sum_{l \in L_{near}} u^{l}_{s_{k'}} \cdot P_{k'l}(\phi_{k'}(\phi(\boldsymbol{x}_{s_k}))), \tilde{\boldsymbol{y}}_{s_k}\Big). \tag{7.12}$$

Then the overall objective of the kth source model is:

$$\mathcal{L}_s = \mathcal{L}_k + \mathcal{L}_{aux}. \tag{7.13}$$

Grouping similar classes to construct fuzzy rules can enrich information of every fuzzy set but degrade the representation of each class. To preserve the identification of each class, source class anchors of each domain are generated to extract present features that highly reflect class information. By this, the anchors can describe classes without referring to the original data. When fuzzy rule number is equal to the class number, which means each class has its individual rule, the clustering prototypes in equation (7.11) will act as source anchors. Otherwise, the averaged mean values of normalized classifier weight vectors are adopted as source anchors. By learning the anchors based on the source private model parameters, source data will not be leaked by decoding these anchors. Thus, employing these anchors does not harm data privacy. The anchor of the cth class is calculated as:

$$\boldsymbol{f}^{c}_{s_k} = \begin{cases} \frac{1}{L_k}\sum_{l=1}^{L_k} \mathrm{Norm}(P_{kl}) \text{ if } L_K \neq C; \\ \frac{\sum_{i=1}^{n_{s_k}} (u^{li}_{s_k})^m \cdot \phi_k(\phi(\boldsymbol{x}^{i}_{s_k}))}{\sum_{i=1}^{n_{s_k}} (u^{li}_{s_k})^m}, \text{ if } L_K = C; \end{cases} \tag{7.14}$$

$$c = l, c = 1, 2, \cdots, C$$

7.3.2 Pseudo Target Label Collection

Given target domain $\mathcal{D}_t = \{\boldsymbol{x}^{j}_{t}\}_{j=1}^{n_t}$, without access to source data, traditional domain adaptation methods relying on matching source and target samples cannot be adopted. To tackle the target task, we employ a pseudo labelling strategy to generate the target model from source models. As source models are available, we feed target data to the kth source model, and select most half nearest rules (denote as L_{near}) to predict target labels. At the very beginning, initializing the target clustering prototypes $\{v^l_{t_k}\}_{k=1}^{L_k}$ as source clustering prototypes $\{v^l_{s_k}\}_{k=1}^{L_k}$, the membership is calculated as:

$$u^l_{t_k} = \frac{1}{\sum_{i=1}^{L_k}\left(\frac{\|\boldsymbol{v}^l_{t_k} - \phi_k(\phi(\boldsymbol{x}_t))\|}{\|\boldsymbol{v}^i_{t_k} - \phi_k(\phi(\boldsymbol{x}_t))\|}\right)^{\frac{2}{m-1}}}. \tag{7.15}$$

The prediction of applying source classifiers is expressed as:

$$\hat{\boldsymbol{y}}_{t_p} = \sum_{l \in L_{near}} u^l_{t_k} \cdot P_{kl}(\phi_k(\phi(\boldsymbol{x}_t))); \tag{7.16}$$

Target prototypes and memberships are then updated alternately with the process of training by fixing the other, which can be expressed as:

$$\begin{aligned} \boldsymbol{v}^l_{t_k} &= \frac{\sum_{i=1}^{n_t}(u^{li}_{t_k})^m \cdot \phi_k(\phi(\boldsymbol{x}^i_t))}{\sum_{i=1}^{n_t}(u^{li}_{t_k})^m}; \\ u^l_{t_k} &= \frac{1}{\sum_{i=1}^{L_k}\left(\frac{\|\boldsymbol{v}^l_{t_k} - \phi_k(\phi(\boldsymbol{x}_t))\|}{\|\boldsymbol{v}^i_{t_k} - \phi_k(\phi(\boldsymbol{x}_t))\|}\right)^{\frac{2}{m-1}}}. \end{aligned} \tag{7.17}$$

The pseudo target labels provided by the pre-trained source model could be noisy due to the data bias between domains. To reduce the label noise, the distillation strategy is designed to collect high-confident target labels assumed to be correct. We call these target samples strong samples. The strong samples are used to further update the pseudo labels of all target samples, which is expected to improve the accuracy of target predictions.

First, denote $\boldsymbol{r} = [r_1, r_2, ..., r_C]$ as the probability vector returned by source classifiers $\{P_{kl}\}_{l=1}^{L_k}$ as in equation (7.16) which indicates the probability of a target sample belonging to the source classes. A threshold a_c of the cth class is defined to identify the potential of a target label being correct. For a pseudo label $\hat{\boldsymbol{y}}_{t_p} = c$, if $r_c \geqslant a_c$, we think this target label is correct with a high probability.

In addition, based on deep clustering (Liang et al., 2020; Caron et al., 2018), we adopt the nearest neighbor to estimate the target labels. Since target data is unlabeled, to estimate the clustering center in each class of target domain, target samples are fed to source classifiers to calculate the probability vectors. The initial clustering center by applying the kth source model then can be written as:

$$\boldsymbol{w}_c^0 = \frac{\sum_{i=1}^{\hat{n}_{t_p}^c} r_c^i \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n}_{t_p}^c} r_c^i}, \tag{7.18}$$

$\hat{n}_{t_p}^c$ is the number of samples in the cth class predicted by equation (7.16). The clustering label of target sample is then estimated by:

$$\hat{\boldsymbol{y}}_{t_d} = \arg\max_{\substack{c \\ \boldsymbol{x}_t \sim \mathcal{D}_t}} \frac{1}{\sum_{i=1}^{C} \left(\frac{\|\boldsymbol{w}_c^0 - \phi_k(\phi(\boldsymbol{x}_t))\|}{\|\boldsymbol{w}_i^0 - \phi_k(\phi(\boldsymbol{x}_t))\|}\right)^2}. \tag{7.19}$$

The target domain is unlabeled with higher data uncertainty, but we hope to collect accurate class information to predict its labels. Thus, the soft class information reflected by probability vector $\boldsymbol{r}$ is transformed into hard class information by replacing the probability vector $\boldsymbol{r}$ with the predicted label $\hat{\boldsymbol{y}}_{t_d}$. This means the initial cluster centers and labels in equations (7.18) and (7.19) are upgraded as:

$$\begin{aligned}
\boldsymbol{w}_c^1 &= \frac{\sum_{i=1}^{\hat{n}_{t_d}^c} \mathbb{1}_{\hat{\boldsymbol{y}}_{t_d}^i = c} \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n}_{t_d}^c} \mathbb{1}_{\hat{\boldsymbol{y}}_{t_d}^i = c}}, \\
\hat{\boldsymbol{y}}_{t_d} &= \arg\max_{\substack{c \\ \boldsymbol{x}_t \sim \mathcal{D}_t}} \frac{1}{\sum_{i=1}^{C} \left(\frac{\|\boldsymbol{w}_c^1 - \phi_k(\phi(\boldsymbol{x}_t))\|}{\|\boldsymbol{w}_i^1 - \phi_k(\phi(\boldsymbol{x}_t))\|}\right)^2}.
\end{aligned} \tag{7.20}$$

$\hat{n}_{t_d}^c$ is the number of cluster samples predicted by equation (7.19).

When $\hat{\boldsymbol{y}}_{t_p} = \hat{\boldsymbol{y}}_{t_d}$ and $r_{\boldsymbol{y}_{t_p} = c} \geqslant a_c$, we select the corresponding target sample as a strong sample. After collecting strong samples, we update the clustering centers in equation (7.20) using the selected target samples and corresponding predicted labels, and then renew the pseudo labels of all target samples, which can be expressed as:

$$\begin{aligned}
\boldsymbol{w}_c^2 &= \frac{\sum_{i=1}^{\hat{n}_{sel}^c} \mathbb{1}_{\hat{\boldsymbol{y}}_{t_p}^i = c} \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{\hat{n}_{sel}^c} \mathbb{1}_{\hat{\boldsymbol{y}}_{t_p}^i = c}}, \\
\hat{\boldsymbol{y}}_{t} &= \arg\max_{\substack{c \\ \boldsymbol{x}_t \sim \mathcal{D}_t}} \frac{1}{\sum_{i=1}^{C} \left(\frac{\|\boldsymbol{w}_c^2 - \phi_k(\phi(\boldsymbol{x}_t))\|}{\|\boldsymbol{w}_i^2 - \phi_k(\phi(\boldsymbol{x}_t))\|}\right)^2}.
\end{aligned} \tag{7.21}$$

n_{sel}^c denotes the number of selected samples in the cth class.

7.3.3 Model Adaptation and Target Task Prediction

When predicting the target task, to fit source models to the target domain, we design a self-supervised strategy to train the target model using the collected pseudo labels. Anchor-based alignment is built to force target data to the source feature spaces.

In freezing source classifiers, only feature extractors are fine-tuned to extract invariant information. When applying the kth fuzzy model, the corresponding generated target model is trained by minimizing the errors between the predictions and the pseudo labels, which is:

$$P_{t_k} = \underset{\substack{\phi_k,\phi \\ \boldsymbol{x}_t \sim \mathcal{D}_t}}{\arg\min} \mathcal{L}(P_{t_k}(\phi_k(\phi(\boldsymbol{x}_t))), \hat{\boldsymbol{y}}_t), \tag{7.22}$$

where

$$\mathcal{L} = -\frac{1}{n_t}\sum_{i=1}^{n_t} \hat{\boldsymbol{y}}_t \log(P_{t_k}(\phi_k(\phi(\boldsymbol{x}_t^i)))). \tag{7.23}$$

P_{t_k} is a linear combination of source classifiers under fuzzy rules, which is:

$$P_{t_k} = \sum_{l \in L_{near}} u_{t_k}^l \cdot P_{kl}. \tag{7.24}$$

$u_{t_k}^l$ is calculated as in equation (7.17).

To reduce the domain shift between source and target domains on the label-level, information maximization loss is employed to parameterize the target outputs being individually certain and globally diverse by encoding the target outputs to one-hot vectors, which is:

$$L_{div}^k = \sum \bar{\boldsymbol{p}}_t \log(\bar{\boldsymbol{p}}_t), \tag{7.25}$$

$\bar{\boldsymbol{p}}_t = \frac{1}{n_t}\sum_{i=1}^{n_t} P_{t_k}(\phi_k(\phi(\boldsymbol{x}_t^i)))$ is a C-dimension vector.

To transform target data to source feature space on the data-level, anchor-based alignment is designed to reduce the data bias. Source anchors $\{f_{s_k}^c\}_{c=1}^C$ are generated from source data as in equation (7.14), which highly represent the class feature information. These anchors will not weaken the data privacy as they are transformations of the original data. Other users (e.g. target domain) cannot estimate source data by decoding the anchors. Target anchors are calculated according to the soft class information returned by applying source fuzzy rules, we still denote it as vector $\boldsymbol{r}$, the target anchor is:

$$\boldsymbol{f}_{t_k}^{c} = \frac{\sum_{i=1}^{n_b} r_c^i \cdot \phi_k(\phi(\boldsymbol{x}_t^i))}{\sum_{i=1}^{n_b} r_c^i}, \tag{7.26}$$

n_b is the batch size. The advantage of calculating target anchors over the batch is that when there is no sample of class c in the randomly selected batch, the anchor can still be generated according to the probabilities of samples from other classes belonging to class c.

The loss function of matching source and target anchors is:

$$\mathcal{L}_{anc}^{k} = \sum_{c=1}^{C} \|\boldsymbol{f}_{t_k}^{c} - \boldsymbol{f}_{s_k}^{c}\|^2. \tag{7.27}$$

The total loss of training for the target model is:

$$\mathcal{L}_t = \sum_{k=1}^{K} (\mathcal{L}(P_{t_k}(\phi_k(\phi(\boldsymbol{x}_t))), \hat{\boldsymbol{y}}_t) + L_{div}^{k} + \mathcal{L}_{anc}^{k}). \tag{7.28}$$

The target label is a mean average combination of the predictions provided by all source classifiers:

$$\boldsymbol{y}_t = \frac{1}{K} \sum_{k=1}^{K} P_{t_k}(\phi_k(\phi(\boldsymbol{x}_t))). \tag{7.29}$$

The processing of the proposed source-free multi-domain adaptation with fuzzy rule-based deep neural networks (SF-FDN) are described in Algorithms 11 and 12.

Algorithm 11 SF-FDN: Source private model training.

```
Input: Source domains;
Define the number of fuzzy rules of each source domain as in equations (7.2);
Initialize cluster prototypes as in equation (7.9);
for $\epsilon = 1, \epsilon < \mathcal{I}_s, \epsilon++,$ do
    Calculate membership as in equation (7.10);
    Update cluster prototypes as in equation (7.11);
    Update the source model as in equation (7.13)
end for
Generate source class anchors as in equation (7.14)
Output: Source models, source anchors.
```

Algorithm 12 SF-FDN: Target model adaptation.

```
Input: Source models, source anchors, target domain;
Initialize target cluster prototypes as source prototypes;
Initialize target memberships as in equation (7.15);
for $\epsilon = 1, \epsilon < \mathcal{I}_t, \epsilon++,$ do
    Calculate pseudo labels predicted by source classifiers as in equation (7.16);
    Update target cluster prototypes and membership as in equation (7.17);
    Calculate pseudo labels predicted by deep clustering as in equation (7.20);
    Collect pseudo target labels as in equation (7.21);
    Generate target class anchors as in equation (7.26);
    Update the target model as in equation (7.28)
end for
Predict target labels as in equation (7.29);
Output: Target labels.
```

7.4 Experiments

In this section, the proposed fuzzy rule-based source-free multi-domain adaptation method is validated on four popular real-world visual datasets, comprising ImageCLEF-DA, Office-31, Office-Caltech10 and Office-Home. All the experiments are classification tasks under the multi-source domain adaptation scenario, both homogeneous and heterogeneous label spaces are applied to validate the proposed method. Classification accuracy is the only criterion used to evaluate the performance.

In the following, section 7.4.1 introduces the datasets, compared methods and parameter settings. Experiment results and analysis are displayed in section 7.4.2. Section 7.4.3 analyzes the generality of the source-only model. Section 7.4.4 analyzes the influence of rule numbers. The ablation study is carried out in section 7.4.5. Section 7.4.7 validates the proposed method under partial and open-set domain adaptation scenarios. Section 7.4.8 displays the data visualization.

7.4.1 Datasets and Baselines

The proposed method is tested on four real-world datasets. Datasets details are listed in Table 7.2.

Office-31 and ImageCLEF-DA include three domains sharing 31 and 12 categories, respectively. For closed-set domain adaptation, three tasks of each dataset can be built: $AW-D$; $AD-W$; $WD-A$ from Office-31 and $I, C \rightarrow P$; $I, P \rightarrow C$; $C, P \rightarrow I$ from ImageCLEF-DA.

Office-Caltech10 and Office-Home contain four domains sharing 10 and 65 categories, respectively. Each of them has four tasks: $A, D, W \rightarrow C$; $C, D, W \rightarrow A$; $A, C, D \rightarrow W$, $A, C, W \rightarrow D$ from Office-Caltech10 and $A, C, P \rightarrow R$; $A, C, R \rightarrow P$; $A, P, R \rightarrow C$; $C, P, R \rightarrow A$ from Office-Home.

Table 7.2 : Classes, domains and samples in experiment datasets.

Dataset\ Classes	Domain\ Samples	Total Samples	Tasks
Office-31\31	Amazon\2817	4110	W,D→A
	Webcam\795		A,D→W
	DSLR\498		A,W→D
ImageCELF-DA\12	Caltech\600	1800	I,P→C
	ImageNet\600		P,C→I
	Pascal\600		I,C→P
Office-Caltech\10	Amazon\958	2533	C,D,W→A
	Webcam\295		A,C,D→W
	DSLR\157		A,C,W→D
	Caltech\1123		A,D,W→C
Office-Home\65	Art\2427	15588	C,P,R→A
	Clipart\4365		A,P,R→C
	Product\4439		A,C,R→P
	RealWorld\4357		A,C,P→R

Domain adaptation with heterogeneous label spaces are validated on Office-Home.

The baselines include related domain adaptation methods with and without source data under both homogeneous and heterogeneous settings which employ the same learning schemes such as self-supervision to adapt the target model. For fair comparison, the baseline methods are trained based on ResNet50. Comparison with single-source methods aims to prove the superiority of learning from multiple domains. Comparison with non-fuzzy methods not only indicates the advantage of fuzzy model, but also shows the superiority of the proposed techniques in data-matching and pseudo label selection. Source data available methods include:

- TransN: Transferable Normalization (Wang et al., 2019a);
- MDD: Margin disparity discrepancy (Zhang et al., 2019b);
- JUMBOT: Joint unbalanced minibatch optimal transport (Fatras et al., 2021);
- RBDA: Reducing bias to source samples (Ye et al., 2021);
- RWOT: Reliable weighted optimal transport (Xu et al., 2020b);
- LtC-MSDA: Learning to combine (Wang et al., 2020a);
- MSCLDA: Learning source contribution for multi-domain adaptation (Li et al., 2021c);
- DCA: Multi-domain adaptation with dynamic classifier alignment (Li et al., 2022a);
- SAN: Partial transfer learning with selective adversarial networks (Cao et al., 2018a);

- ETN: Learning to transfer examples for partial domain adaptation (Cao et al., 2019);
- SAFN: Adaptive feature norm approach for unsupervised domain adaptation (Xu et al., 2019b);
- DARL: Domain adversarial reinforcement learning for partial domain adaptation (Chen et al., 2022b);
- MSAN: Attention guided for partial domain adaptation (Zhang and Zhao, 2021);
- OSBP: Open set domain adaptation by backpropagation (Saito et al., 2018);
- STA: Separate to adapt: Open set domain adaptation via progressive separation (Liu et al., 2019a);
- DAOD: Open set domain adaptation: theoretical bound and algorithm (Fang et al., 2021);
- LtGUR: Learning to generate the unknowns as a remedy for open-set adaptation (Baktashmotlagh et al., 2022).

Source free methods include:

- BAIT: Domain adaptation without source data by casting a bait (Yang et al., 2021b);
- PrDA: Progressive domain adaptation (Kim et al., 2020);
- SHOT: Source hypothesis transfer with information maximization (Liang et al., 2020);
- SDDA: Source data free domain adaptation- domain impression (Kurmi et al., 2021b);

- G-SFDA: Generalized source-free domain adaptation (Yang et al., 2021c);
- AAN: Adaptive adversarial network (Xia et al., 2021);
- NRC: Intrinsic neighborhood structure for source-free domain adaptation (Yang et al., 2021a);
- JNUSF: Source-free domain adaptation with Jacobian Norm (Li et al., 2022b);
- CDCL: Cross-domain contrastive learning for unsupervised domain adaptation (Wang et al., 2022);
- PGL: Source-free progressive graph learning for open-set domain adaptation (Luo et al., 2022).

ETN, SAFN, DARL and MSAN are compared under a partial domain adaptation setting, while OSBP, STA, DAOD, LtGUR and PGL are compared under an open-set domain adaptation setting. All compared results are collected from previous publications. For single source-free domain adaptation methods, we take the average predictions from all source domains as the multi-source results.

$ResNet50$ is employed as the backbone complemented by Pytorch. Parameters are updated based on back-propagation with Stochastic Gradient Descent (SGD), the momentum is 0.9, the learning rate η follows the same strategy in (Ganin and Lempitsky, 2015), which is $\eta = \frac{\eta_0}{(1+10\epsilon)^{0.75}}$, where $\eta_0 = 0.01$, ϵ is the training progress changing linearly from 0 to 1. The learning rate of the shared network is one tenth of other layers. Batch size $n_b = 64$, the smoothing parameter $\mu = 0.1$. Threshold a_c is defined as medium value of the predicted probabilities in each category. For datasets ImageCLEF-DA and Office-Caltech10, we set the rule numbers as their class numbers. For datasets Office-31 and Office-Home, rules numbers are defined by their correlation coefficient among classes. The value of threshold a_ρ is affected by the sample number in each domain. Domains containing few samples applies a small value while domains containing a large number of samples take on a greater value. For dataset Office-31, ImageCLEF-DA, Office-Caltech10, the value of a_ρ is between $[0.4, 0.5]$, for Office-Home, the value is between $[0.45, 0.65]$. Domains A, D and W have 15, 16 and 15 rules respectively, domains A, C, P and R have 5, 6, 7 and 7 rules respectively. Under heterogeneous label space settings on dataset Office-Home, for partial domain adaptation, domains A, C, P and R have 5, 6, 7 and 7 rules respectively, while for open-set domain adaptation, domains A, C, P and R have 5, 6, 8 and 8 rules respectively.

7.4.2 Results and Analysis

Tables 7.3, 7.4, 7.5 and 7.6 show the results of the proposed method and the baselines under closed-set domain adaptation. We compare the proposed method SF-FDN with a fuzzy rule-based baseline MDAFuz, and other non-fuzzy baselines. It indicates that the proposed method performs the best on most tasks and achieve the highest average performance on four datasets.

To compare the proposed SF-FDN method with the source-free domain adaptation methods, the average accuracy is improved by 1.5% on dataset Office-31, 0.6% on dataset ImageCLEF-DA, 0.5% on dataset Office-Caltech10 and 1.3% on dataset Office-Home, respectively. This indicates that introducing a fuzzy system to handle soft information among samples from different categories can leverage richer transfer knowledge across domains. The proposed SF-FDN improves the average accuracy of baselines with source data by 0.4% on dataset Office-31, 0.3% on dataset Office-Caltech10 and 0.8% on dataset Office-Home compared with the latest domain adaptation method with source data. On dataset ImageCLEF-DA, the proposed SF-FDN and baseline MDAFuz, another method based on fuzzy rules, achieve the same average performance. It means extracting soft information is more suitable for this dataset. Even though the two methods gain the same average performance, we deal with domain adaptation under a more difficult setting, and obtain a superior performance on most tasks in this dataset. It also shows that, except for the proposed method, several source-free methods based on pseudo-labelling, such as SHOT and BAIT, produce a similar performance compared with other methods with source data, meaning that self-supervised training is advantageous in taking usable information in the target domain to help the transfer.

7.4.3 Generality Analysis of Source Private Model

When handling source-free domain adaptation employing a self-training strategy. there are two main questions: how to improve the generality of the source only model and how to collect pseudo labels with low noise. In this section, we expect fuzzy rules to have the superiority to take full use of class information to learn classifiers with high cross-domain ability, and design auxiliary tasks to enhance the generality of source-only models. The high cross-domain performance of the source model is beneficial to collecting low noisy pseudo target labels at the very beginning. This section conducts experiments on a source-only model for analyzing the performance of the proposed enhancement strategy.

Taking datasets Office-31 and Office-Home as examples, Tables 7.7 and 7.8 show the performance of source-only models trained without fuzzy rules and auxiliary tasks. The results are returned by applying source models on the target domain directly without fine-tuning, which indicates the ability across tasks of source models. Method "S" represents predictions of single source domain, "M" indicates the performance of multi-source domains. "Non-fuzzy" means the models are trained with auxiliary tasks but without fuzzy rules, "Non-auxiliary" means training with fuzzy rules but without auxiliary tasks, "Proposed" means both fuzzy rules and auxiliary tasks are used.

It can be seen that the multi-source model outperforms the single source model,

Table 7.3 : Comparison (%) of the proposed fuzzy rule-based deep network and the baselines on dataset Office-31

Standards	Method	A,W→D	A,D→W	W,D→A	Avg
Source data	ResNet	99.3	96.7	62.5	86.2
	TransN	97	97.2	73.8	89.3
	RBDA	**100.0**	99.0	74.2	91.1
	MDD	96.8	96.6	73.4	88.9
	RWOT	97.3	97.3	**77.7**	90.8
	MSCLDA	99.8	98.8	73.7	90.8
	MDAFuz	99.7	99.0	74.0	90.9
	DCA	99.6	98.9	75.1	91.2
Source free	ResNet	97.5	95.4	60.2	84.4
	BAIT	98.8	98.5	71.1	89.5
	PrDA	96.7	93.8	73.2	87.9
	SHOT	94.9	97.8	75.0	89.2
	SDDA	99.8	99.0	67.7	88.8
	AAN	97.3	96.6	76.1	90.1
	NRC	97.9	94.9	75.2	89.3
	JNUSF	97.9	95.5	76.4	89.9
	CDCL	97.2	95.3	75.3	89.3
	SF-FDN	**100.0**	**99.2**	75.8	**91.6**

Table 7.4 : Comparison (%) of the proposed fuzzy rule-based deep network and the baselines on dataset ImageCLEF-DA

Standards	Method	I,C→P	I,P→C	P,C→I	Avg
Source data	ResNet	74.8	91.5	83.9	83.4
	RBDA	78.5	**98.0**	91.4	89.3
	RWOT	80.2	97.3	92.8	**90.1**
	MSCLDA	79.5	95.9	94.3	89.9
	MDAFuz	79.4	96.3	**94.5**	**90.1**
	DCA	78.9	96.2	93.9	89.7
Source free	SHOT	79.2	96.2	93.2	89.5
	SF-FDN	**80.2**	97.3	92.7	**90.1**

Table 7.5 : Comparison (%) of the proposed fuzzy rule-based deep network and the baselines on dataset Office-Caltech10

Standards	Method	A,D,W→C	C,D,W→A	A,C,D→W	A,C,W→D	Avg
Source data	ResNet	82.5	91.2	98.9	99.2	93.0
	MSCLDA	94.1	95.3	99.1	98.5	96.8
	DCA	94.7	96.0	99.7	99.1	97.4
Source free	ResNet	92.1	96.3	98.0	99.5	96.5
	BAIT	95.7	**97.5**	98.0	97.5	97.2
	PrDA	94.6	97.3	97.6	97.1	96.7
	SHOT	**95.8**	95.7	99.6	96.8	97.0
	SF-FDN	94.9	95.9	**100.0**	**100.0**	**97.7**

Table 7.6 : Comparison (%) of the proposed fuzzy rule-based deep network and the baselines on dataset Office-Home

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Source data	ResNet	67.8	71.3	51.8	53.4	61.1
	TransN	76.7	75.7	54.1	64.1	67.6
	MDD	75.9	75.8	56.2	64.6	68.1
	RWOT	77.3	75.8	52.8	64.5	67.7
	JUMBOT	78.3	77.8	55.9	67.0	70.0
	MSCLDA	80.6	79.9	61.4	71.6	73.4
	LtC-MSDA	80.1	79.2	**64.1**	67.4	72.7
	DCA	81.4	80.5	63.6	72.1	74.4
Source free	ResNet	76.3	78.8	50.1	50.9	64.0
	BAIT	77.2	79.4	59.6	71.1	71.8
	PrDA	76.8	79.1	57.5	69.3	70.7
	SHOT	81.5	83.0	57.2	72.1	73.5
	G-SFDA	82.2	83.4	57.9	72.0	73.9
	AAN	81.4	81.1	58.4	69.9	73.9
	NRC	81.2	81.9	57.6	68.1	72.2
	JNUSF	81.2	81.1	56.8	70.8	72.5
	SF-FDN	82.7	**83.7**	60.7	**73.7**	**75.2**

indicating that enriching source knowledge is helpful in learning a classifier that can perform on multiple domains. The proposed method performs best on two datasets compared with models trained without fuzzy rules or auxiliary tasks. For a dataset with fewer categories and samples (Office-31), models trained with fuzzy rules but without auxiliary tasks achieve greater accuracy than those trained with auxiliary tasks but without fuzzy rules. It means fuzzy rules have a significant advantage to improve the generality of source models. While for a dataset with more categories and samples (Office-Home), auxiliary learning is more important to leveraging source knowledge as the models trained with auxiliary tasks outperform those without. This is because Office-Home contains more domains than Office-31, and some domains with low relatedness are learned together, degrading the performance of classifiers since a classifier fitting all domains well may not exist. This encourages us to explore which tasks should be learned together to reduce the negative transfer and improve future positive transfer.

7.4.4 Influence of Rule Numbers

To explore how the number of rules affects the performance of the transfer, this section shows the results of the proposed method trained with different rule numbers. Tables 7.9 and 7.10 show the classification results of the proposed fuzzy rule-based method with different rule numbers on datasets Office-31 and Office-Home. Standard “non” indicates only one rule is used, which can be treated as non-fuzzy classification. Standard “$C/2$” means the rule number is half of the class number, while “C” means the rule number is set as a class number. The results show that rule numbers defined by the proposed grouping strategy based on correlation coefficient achieve the highest performance on target domain. It indicates that too few or too many rules can result in degradation of the transfer.

Fewer rules might fail to discover the specific information among different classes.

Table 7.7 : Accuracy (%) on dataset Office-31 of source only models.

Standards	Method	A,W→D	A,D→W	W,D→A	Avg
Non-fuzzy	S	98.6	96.2	64.7	86.5
		98.6	96.2	64.5	
	M	98.8	96.4	65.8	87.0
Non-auxiliary	S	99.0	95.2	63.5	86.6
		99.6	97.2	65.0	
	M	99.4	**97.4**	66.0	87.6
Proposed	S	99.6	95.2	65.0	87.1
		99.8	97.2	65.9	
	M	**99.8**	**97.4**	**66.4**	**87.9**

Setting the rule number as a value equal to or greater than the number of classes should be advantageous in extracting class information and learning high-performance classifiers, but experiments reveal different assumptions on datasets containing many categories. We think this is caused by the data unbalance of categories and domains. For a classifier under a rule that contains fewer highly representative samples, the learning may fail to provide correct predictions because other class samples occupying a large proportion will dominate the training. Besides, having too many rules requires a high computing environment such as computer memory and compute units. Thus, defining appropriate fuzzy rules is beneficial in overcoming the problems mentioned above.

Table 7.8 : Accuracy (%) on dataset Office-Home of source only models.

Standards	Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Non-fuzzy	S	81.5	76.9	51.7	68.2	69.7
		80.8	77.3	52.2	68.2	
		81.1	78.5	51.4	68.8	
	M	**81.6**	78.1	52.0	**69.6**	70.3
Non-auxiliary	S	76.0	71.7	49.7	61.4	65.1
		73.5	72.2	47.2	60.0	
		75.3	77.2	50.4	66.1	
	M	79.7	76.9	53.1	66.7	69.1
Proposed	S	80.4	76.5	53.5	67.2	69.8
		79.5	77.6	54.0	68.2	
		80.6	**78.7**	53.1	67.8	
	M	80.7	**78.7**	**54.1**	68.7	**70.6**

Table 7.9 : Accuracy (%) on dataset Office-31 of ablation study.

Standards	A,W→D	A,D→W	W,D→A	Avg
non	99.5	98.7	75.5	91.3
C	99.8	98.7	75.3	91.2
Proposed	**100.0**	**99.2**	**75.8**	**91.6**

Table 7.10 : Accuracy (%) on dataset Office-Home of ablation study.

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
non	**83.1**	83.1	60.1	73.3	74.9
C/2	82.0	82.4	**61.1**	72.6	74.5
C	82.8	79.7	56.5	66.0	71.3
Proposed	82.7	**83.7**	60.7	**73.7**	**75.2**

7.4.5 Ablation Study

To validate the performance of different modules used in the proposed method, Tables 7.12 and 7.13 display the ablation study on datasets Office-31 and Office-Home. Three modules affect the training of the target model:(1) selecting strong target samples to predict low noisy pseudo labels (denote as $\mathcal{L}_{sel}$); (2) balancing domain shift using information maximization loss (reflected by $\mathcal{L}_{div}$), and (3) forcing target data to source feature space using anchor-based alignment (reflected by $\mathcal{L}_{anc}$), the setting of ablation study is detailed in Tabel 7.11, "×" means training without the module, while "✓" means training with the module.

It indicates that information maximization loss L_{div} is more important than other modules as the model trained without it produces a lower performance on both datasets. Target model of dataset Office-Home trained without sample selection L_{sel} is inferior to that trained without anchor-based alignment L_{anc}, while on dataset Office-31, the situation is different. It means sample selection mainly affects the model on the dataset with a large number of samples and categories, while a dataset with fewer samples relies more on anchor-based alignment. The difference is due to the quality of source anchors. Even in supervised learning, predicting a dataset containing a large number of samples and categories is more difficult than in a

Table 7.11 : Setting of ablation study.

Standards	Target sample selection	Information maximization loss	Anchor-based alignment
L_{div}	✓	×	✓
L_{anc}	✓	✓	×
L_{sel}	×	✓	✓
Proposed	✓	✓	✓

dataset with fewer categories. Thus, the anchors from a small dataset can describe class information more accurately than those from a large dataset.

Table 7.12 : Accuracy (%) on dataset Office-31 of ablation study.

Standards	A,W→D	A,D→W	W,D→A	Avg
$\mathcal{L}_{div}$	99.9	98.4	73.1	90.5
$\mathcal{L}_{anc}$	99.6	98.4	72.9	90.3
$\mathcal{L}_{sel}$	99.9	98.6	74.8	91.1
Proposed	**100.0**	**99.2**	**75.8**	**91.6**

7.4.6 Trade-off Parameter Sensitivity Analysis

This section analyzes the sensitivity of trade-off parameter. Trade-off parameters control the contribution level of auxiliary task $\mathcal{L}_{aux}$, information maximization loss $\mathcal{L}_{div}$ and anchor-based alignment $\mathcal{L}_{anc}$. α ($\mathcal{L}_{aux}$) and β ($\mathcal{L}_{div}$) are experience values in previous domain adaptation methods (Liang et al., 2020; Li et al., 2021c). Here

Table 7.13 : Accuracy (%) on dataset Office-Home of ablation study.

Standards	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
$\mathcal{L}_{div}$	82.0	81.8	55.4	70.6	72.4
$\mathcal{L}_{anc}$	**82.8**	**83.7**	60.3	73.0	75.0
$\mathcal{L}_{sel}$	**82.8**	**83.7**	60.4	73.3	75.1
Proposed	82.7	**83.7**	**60.7**	**73.7**	**75.2**

we provide the experiment on λ, which refects the term $\mathcal{L}_{anc}$. Taking datasets Office-31 and Office-Home as examples, the results are shown in Tables 7.14 and 7.15. It can been seen that when $\lambda = 0.5$, the proposed method achieves the highest performance.

Table 7.14 : Accuray (%) on dataset Office-31 with different values of parameter λ.

λ	A, W→D	A, D→W	W, D→A	Avg
0.3	99.8	98.5	75.2	91.2
0.5	**100.0**	**99.2**	**75.8**	**91.6**
0.7	99.9	98.4	75.3	91.2
1	99.8	98.6	75.2	91.2

7.4.7 Validation under Heterogeneous Label Space Setting

This section describes the experiments on the dataset Office-Home under heterogeneous label spaces settings, including partial and open-set domain adaptation. In partial domain adaptation where target label space is a proper subset of source label space, we choose 25 classes as target domain, and source domains include all classes.

Table 7.15 : Accuray (%) on dataset Office-Home with different values of parameter λ.

λ	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
0.3	82.0	83.6	60.4	73.0	74.8
0.5	**82.7**	**83.7**	**60.7**	**73.7**	**75.2**
0.7	82.6	82.9	60.8	73.5	75.0
1	82.1	83.6	60.6	73.2	74.9

In open-set domain adaptation, source label space is contained inside target label space. We select 25 classes to build source domains while target domain includes all classes. The results are shown in Tables 7.16 and 7.17.

Compared with non-fuzzy baselines, the proposed method achieves the highest accuracy under both partial and open-set scenarios, meaning the proposed method based on fuzzy rules has superiority over other methods. In addition, it indicates that the target model generated from multiple source domains with joint training takes advantage of similar tasks to improve the transfer across domains.

7.4.8 Visualization Analysis

Taking task $WD - A$ from dataset Office-31 as an example, Fig. 7.2 shows the target data in classification space before and after adapting source models under the closed-set domain adaptation setting using T-SNE visualization (Maaten and Hinton, 2008). Categories are shown in different colors. It can be seen that, before adaptation, features from different classes are mixed, while the proposed method divides target samples with clear boundaries.

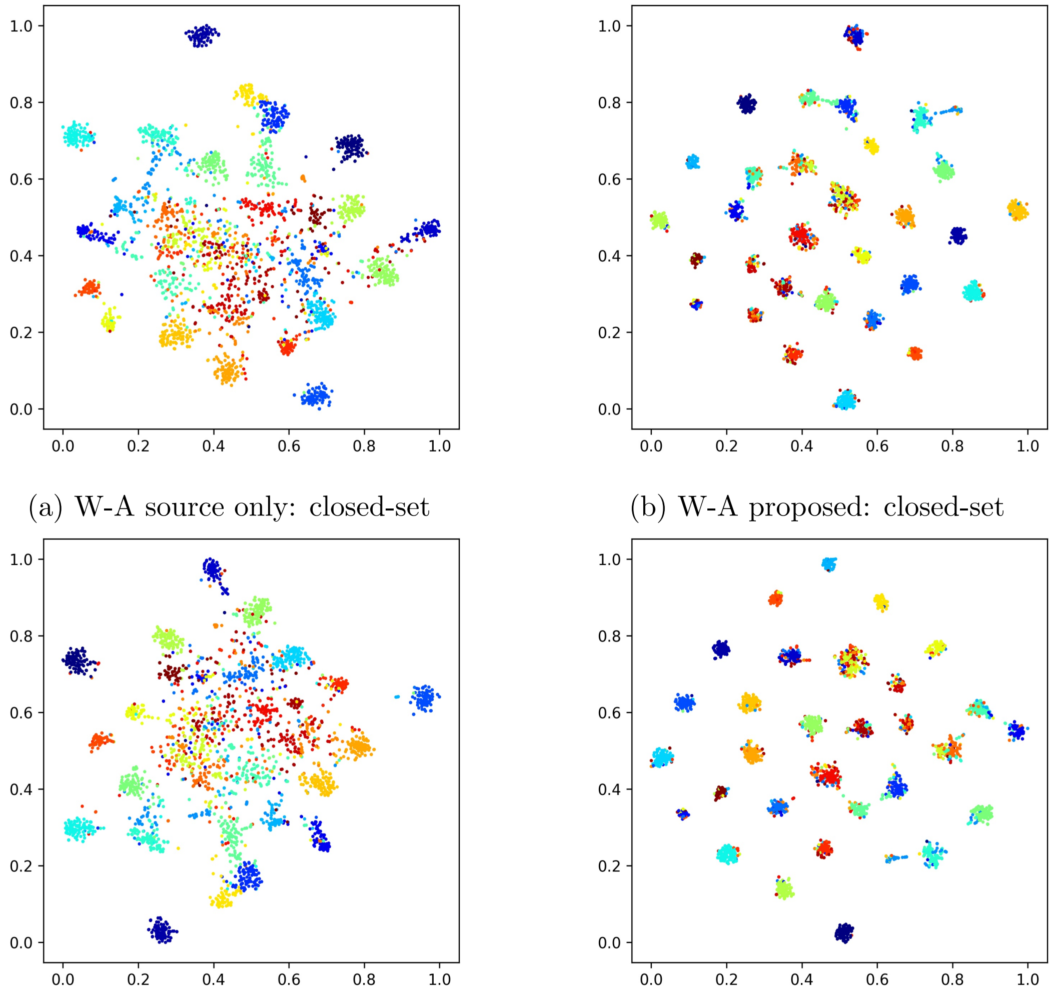
Fig. 7.3 shows the data visualization of target domain A from Office-Home

Table 7.16 : Comparison (%) of the proposed fuzzy rule-based deep network and the baselines on dataset Office-Home under partial domain adaptation.

Standards	Method	A,C,P→R(25)	A,C,R→P(25)	A,P,R→C(25)	C,P,R→A(25)	Avg
Source data	ResNet	71.2	67.2	45.4	61.6	61.4
	SAN	77.5	70.8	46.4	66.5	65.3
	ETN	79.6	75.7	57.4	69.0	70.4
	SAFN	79.9	76.4	58.2	72.9	71.9
	DARL	84.2	77.5	54.5	72.0	72.1
	JUMBOT	83.3	78.2	63.3	77.0	75.5
	MSAN	80.4	76.2	56.7	67.2	70.1
Source free	SHOT	**88.4**	82.4	64.0	77.6	78.1
	SF-FDN	88.3	**83.3**	**66.3**	78.2	**79.0**

Table 7.17 : Comparison (%) of the proposed fuzzy rule-based deep network and the baselines on dataset Office-Home under open-set domain adaptation.

Standards	Method	A,C,P(25)→R	A,C,R(25)→P	A,P,R(25)→C	C,P,R(25)→A	Avg
Source data	ResNet	65.3	63.1	62.8	69.8	65.3
	OSBP	64.2	62.8	65.4	70.2	65.7
	STA	70.4	67.7	66.4	73.3	69.5
	DAOD	79.8	73.5	58.6	67.2	69.8
	LGUR	82.3	78.5	58.8	71.3	72.7
Source free	SHOT	82.4	79.3	61.4	64.5	71.9
	PGL	**86.1**	79.2	63.8	**75.1**	76.1
	SF-FDN	83.1	**83.5**	**66.5**	73.2	**76.6**



(a) W-A source only: closed-set

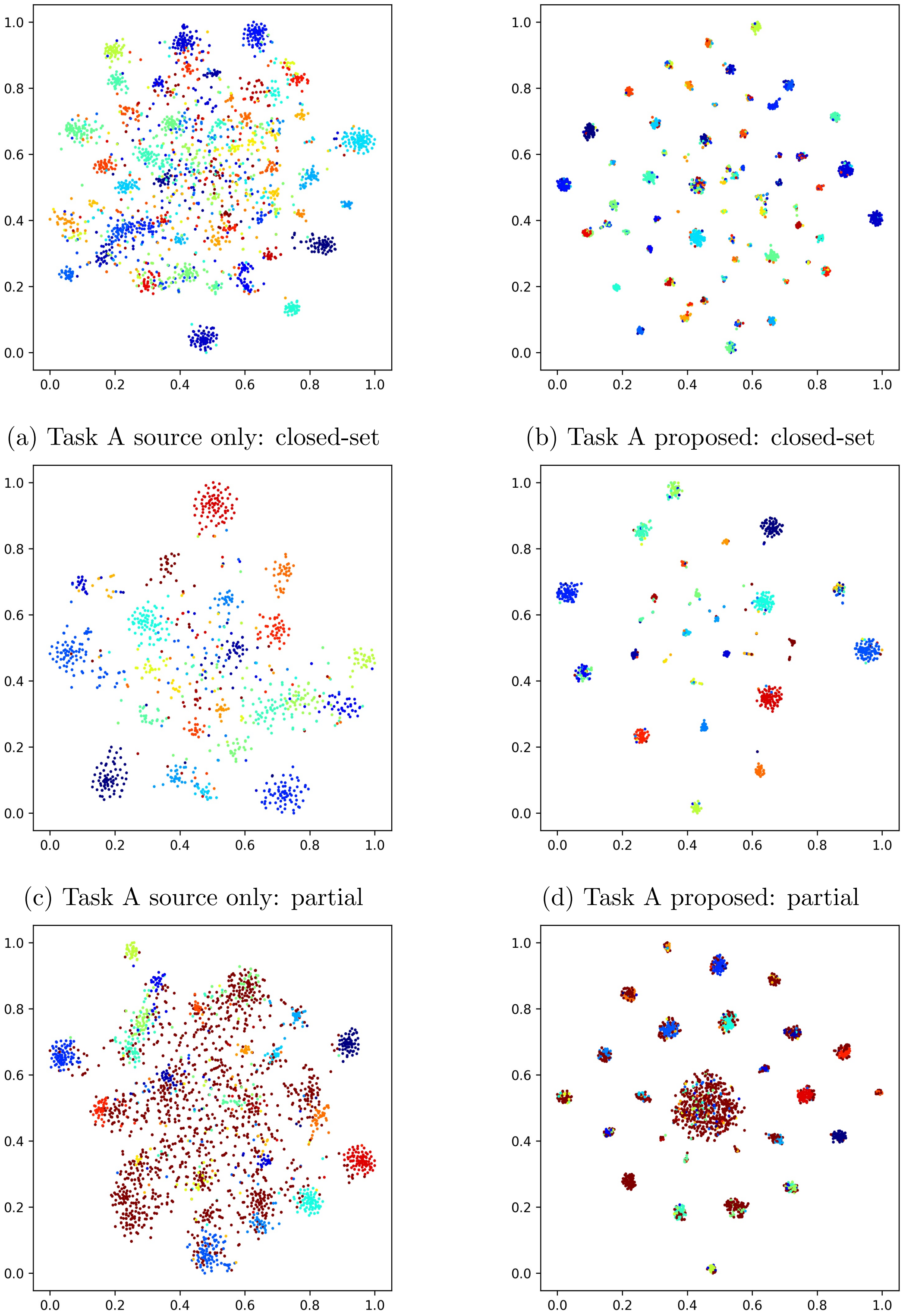
(b) W-A proposed: closed-set

(c) D-A source only: closed-set

(d) D-A proposed: closed-set

Figure 7.2 : T-SNE visualization on target domain A from dataset Office-31.

under both homogeneous and heterogeneous domain adaptation settings. We can see target classes distinctly separate from each other distinctly after adaptation. In partial domain adaptation, the target domain contains 25 classes, and quiet large distances can be seen within each pair of classes. In open-set domain adaptation, there are 65 classes in the target domain. However, source classifiers can only identify 25 classes, the other unshared 40 classes in the target domain are treated as unknown classes (samples in very deep red color). We can see the unknown classes are grouped after fine-tuning, while the share classes are divided clearly into different classes.



(a) Task A source only: closed-set

(b) Task A proposed: closed-set

(c) Task A source only: partial

(d) Task A proposed: partial

(e) Task A source only: open-set

(f) Task A proposed: open-set

Figure 7.3 : T-SNE visualization on target domain A from dataset Office-Home.

7.5 Summary

This chapter proposes a fuzzy rule-based deep structure for source-free multi-domain adaptation. It is an early study of fuzzy domain adaptation without source data. The proposed method introduces the Takagi-Sugeno fuzzy model to source-free domain adaptation. It defines source fuzzy rule numbers by grouping similar classes into the same cluster according to the correlation coefficient within each pair of classes. To train source models with high generality, which is of advantage in predicting target labels with low noise at the beginning, auxiliary tasks are designed by jointly training fuzzy rules from other source domains. The auxiliary training strategy shares source parameters without referring to the original data from other domains, which can protect data privacy. To collect high confident target pseudo labels, a samples selection strategy is built by combining the predictions of source classifiers and deep clustering. Experiments on real-world datasets validate the superiority of the proposed method. The proposed method based on fuzzy rules results in higher performance than baselines trained with and without source data.

Some questions remain unsolved. For example, multiple source domains have their individual models, which requires a large memory of computer memory due to the sizeble number of parameters needed during training and transferring. Furthermore, similarities among each pair of source and target domains are different, meaning multiple source domains contribute to target domain differently. However, without access to source data, it is difficult to measure the similarities between source and target domains. In the future studies, we will try to solve the problems we have illustrated to reduce the computing complexity and learn the contribution of source domains in improving the transfer performance.

Chapter 8

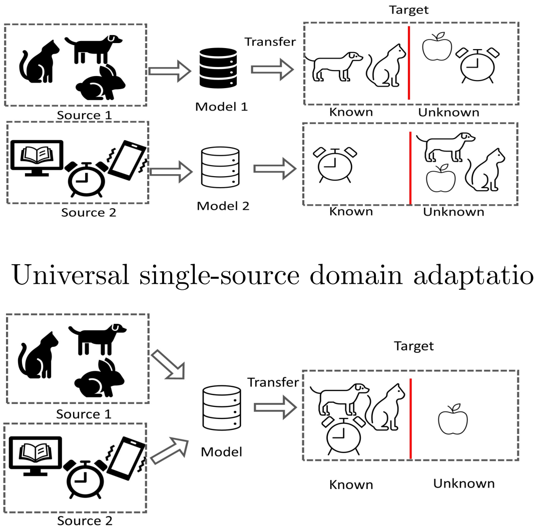
Unified Learning for Multi-Source-Free Universal Domain Adaptation

8.1 Introduction

In previous chapters, we deal with transfer learning under homogeneous scenario, where the source and target domains have the same label spaces (Yue et al., 2021; Chen et al., 2022c; Huang et al., 2022). However, this condition cannot always be satisfied in real world applications. Source and target domains containing different categories are more common. Thus, transfer learning tackling heterogeneous label spaces is developed, including partial (Cao et al., 2018a; Zhang and Zhao, 2021), open-set (Saito et al., 2018; Xu et al., 2021b) and universal settings (Li et al., 2021a). Partial transfer learning deals with knowledge transfer across domains where the source label space is larger than that of the target domain (Cao et al., 2019). A general solution for selecting unshared source samples in partial domain adaptation is based on the relevance between source samples and the target domain. Samples gaining low relevance are regarded as outliers from unshared classes. By removing these samples during adaptation, it is expected to reduce the influence of unshared classes when transferring source model to the target domain. (Chen et al., 2022b). Open-set transfer learning is designed to handle transfer learning where the target domain contains more categories than the source domain (Fang et al., 2021). It has to classify known classes (classes shared by source and target domains) and unknown classes (target private classes). To detect unknown samples, hard rejection and soft rejection based on a threshold defined by clustering or entropy assumption are developed (Jing et al., 2021; Xu et al., 2021b).

Universal transfer learning handles a more challenging scenario where source and target domains have private categories respectively (You et al., 2019). Compared with partial and open-set domain adaptation, universal domain adaptation has to classify known classes without introducing too much unrelated information of source private classes, and distinguish target unknown classes simultaneously. Combining relevance measurement and entropy assumption is a popular method to identify known classes and unknown classes (Saito et al., 2020). Most existing universal domain adaptation methods rely on access to the source data to achieve transfer across domains. However, source data is not always available due to privacy issues, especially in real applications. For example, in healthcare, patient information like disease history is very private information and cannot be public or shared. Besides, there can be multiple source domains for a target domain. Transferring information from multi-source domains and the label heterogeneity issue among multi-source domains remain unsolved.

To address universal transfer learning without source data, encouraged by source-free domain adaptation, data generation is employed to generated source data, including positive and negative samples (Kundu et al., 2020a). Positive samples are used to adapt source and target data while negative samples are used to train unknown classifier. However, these method requires a very large number of generated data, which can be space consuming. In addition, when there are multiple source domains, existing methods cannot be applied efficiently, especially when the source domains have different label spaces. As shown in Fig. 8.1(a), given multi-source domains, existing universal domain adaptation methods transfer knowledge from a single source domain to the target domain. This requires training individual source model in every source domain, the individual source model can only classify samples from the classes shared by the corresponding source and target domains. For example, source model 1 can classify Dog and Cat, while source model 2 can classify Clock. Training individual model requires learning more parameters and the individual model cannot classify all known classes. That is why we propose universal multi-source-free domain adaptation. As shown in Fig. 8.1(b), the purpose of handling multi-source universal domain adaptation is to combine knowledge from the source domains to identify more transferable information to assist in the target domain. If the model can tackle source domains with heterogeneous label spaces, it has the ability to handle multi-source domains with the same label space.



(a) Universal single-source domain adaptation

(b) Universal multi-source domain adaptation

Figure 8.1 : Universal domain adaptation with single and multiple source domains.

In this chapter, we propose a unified learning model for universal multi-source-free domain adaptation. The proposed method learns one model to predict multiple tasks from both source and target domains. The unified learning model combines source invariant information that can be transferred among domains to predict target known classes. At the same time, it designs a source category classifier to assist in generating source class anchors which are employed to match target data to source categories. The source category discriminator can guarantee the flexibility of the proposed model to handle multi-source domains with homogeneous and heterogeneous label spaces. Both target unknown classes and source private classes are identified when pseudo-labeling target samples, which is expected to improve the classification performance on known and unknown classes. Our contributions are summarized as follows:

- We propose a unified learning model to learn multiple tasks for universal source-free domain adaptation. Compared with most existing universal adaptation methods, it has the ability to handle multi-domains without accessing or sharing source data. Besides, different from many multi-domain adaptation methods, the proposed model avoids the need to train an independent model for each source domain which can reduce the number of parameters.

- We propose a learning scheme that is flexible for source domains with homogeneous and heterogeneous label spaces. Label heterogeneity among multiple source domains is rarely considered in domain adaptation methods with and without source data. In the proposed method, both homogeneous and heterogeneous source label spaces are explored to transfer knowledge to the target domain.

- We design an anchor-based clustering strategy to adapt target data to the source domain. A generation function is built to create source anchors based on contrastive learning. The adaptation considers not only target unknown classes, but also source private classes which many existing methods ignored to guarantee the ability of the proposed model to classify known and unknown classes.

8.2 Problem Setting and Notations

We focus on universal multi-source-free domain adaptation, where source domains with both homogeneous and heterogeneous label spaces are considered. Table 8.1 describes the symbols used in this chapter.

Table 8.1 : Notations and descriptions.

Notation	Description
$\mathcal{D}_{s_k}$, $\mathcal{D}_t$	source/target domain, k is source index
$\mathcal{C}_{s_k}$, $\mathcal{C}_t$	source/target label space
$\boldsymbol{x}_{s_k}$, $\boldsymbol{x}_t$	source/target data
$\boldsymbol{y}_{s_k}$	label of $\boldsymbol{x}_{s_k}$
$\hat{\boldsymbol{x}}_s$	generated source sample
$\hat{\boldsymbol{v}}_s$	generated source center
$\boldsymbol{v}_t$	clustering prototype from the target domain
ϕ	feature extractor
G	source sample generator
P	unified learning classifier
P_c	source category discriminator

8.3 The Proposed Unified Learning Model for Universal Multi-Source-Free Domain Adaptation

The proposed method is illustrated in Fig. 8.2. Fig. 8.2(a) indicates the source model training. The unified learning model is trained on multi-source domains by combining the feedback from source domains without sharing source data. To provide source data information for adapting the source model to the target domain without accessing the source data, taking the source ground-truth labels as inputs, a generator is built to create source-like samples. Considering multi-source domains can have different label spaces, to guarantee that the generator can handle the source heterogeneity, source category discriminator is designed for each source domain to provide constraint for its own source classes and reduce the influence of unshared source classes. Fig. 8.2(b) is the procedure of model adaptation in the target domain. Intent to perform the source model on target task, classifier layers are frozen while the backbone is fine-tuned under the supervision of self-training and data-matching. To match source and target data, generated source anchors are adopted to reduce their distance. To self-supervise the training, source private and target unknown classes are identified to reduce pseudo label noise and collect high confident target labels.

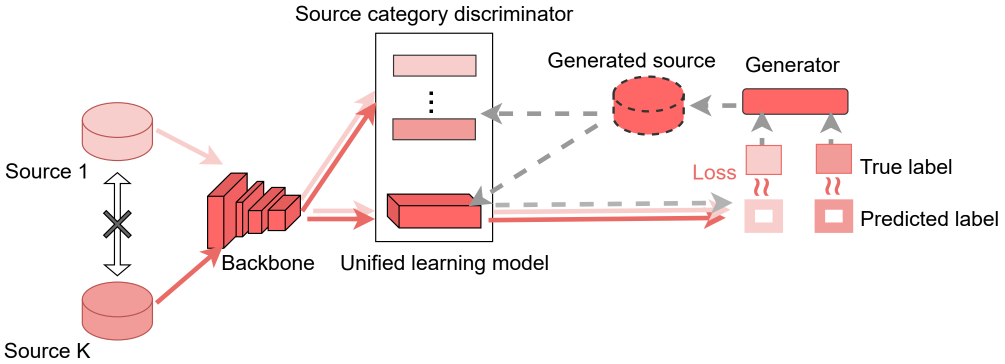
8.3.1 Unified Learning Model Training

In this section, to avoid training an independent source model in each source domain which requires more computation and generates a large number of parameters, we propose a unified learning model to handle multiple source tasks. It is important to guarantee that the unified learning model can be performed on multiple source domains with both homogeneous and heterogeneous label spaces. To achieve this, denote the source label space as $\mathcal{C}_s = \mathcal{C}_{s_1} \cup \cdots \cup \mathcal{C}_{s_k}$, a classifier $P \in \mathbb{R}^{C_s}$ is trained based on multiple source domains, where C_s indicates C_s-dimension. The classifier P can be trained by minimizing the error between the outputs and the ground-truth labels with regard to the risk minimization principle (Vapnik and Vapnik, 1998), which is expressed as:

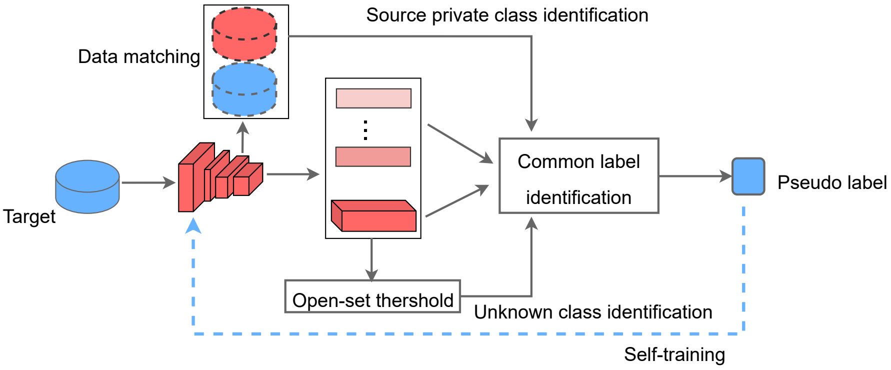
$$\begin{aligned} P = & \underset{(\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \in \mathcal{D}_{s_k}}{\underset{P}{\arg\min}} \; L(P(\phi(\boldsymbol{x}_{s_k})), \boldsymbol{y}_{s_k}), \\ & k = 1, \cdots, K. \end{aligned} \tag{8.1}$$

L is cross-entropy loss:

$$\begin{aligned} L = & -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(P(\phi(\boldsymbol{x}_{s_k}^i))), \\ & k = 1, \cdots, K. \end{aligned} \tag{8.2}$$



(a) Unified source model training. Unified model is trained on multi-source domains without sharing data. Source category discriminator is built to assist in source data generating which can handle source heterogeneity.


Source private class identification
Data matching
Common label identification
Pseudo label
Target
Open-set thershold
Unknown class identification
Self-training

(b) Target model adaptation. Data matching is designed based on the generated source-like data to reduce data shift. Source private and target unknown classes are identified to collect high confident target labels to provide self-supervision.

Figure 8.2 : The procedure of the proposed method.

8.3.2 Source Data Generation

To adapt source and target domains without accessing the source data, we generate source-like data to match the source and target distributions. Considering that generating a large number of source-like samples can occupy a large amount of computer memory, here we generate the source class anchors which represent the source information to reduce the amount of memory being used. The generated source anchors should satisfy two conditions: one is that the class anchors are assumed to be classified to its corresponding categories, another is that the source samples from the same class should be close to the corresponding anchors. Denote the source label as $\boldsymbol{y}_s \in \mathcal{C}_s$, the generation of source data can be expressed as:

$$\hat{\boldsymbol{x}}_s = G(\boldsymbol{y}_s) \tag{8.3}$$

For the first condition, following the work in (Qiu et al., 2021), source data generator G is optimized by minimizing the cross-entropy loss between the classification predictions of the generated anchors and their labels, which is:

$$G = \arg\min_{G} L(P(\phi(G(\boldsymbol{y}_s))), \boldsymbol{y}_s), \tag{8.4}$$

where

$$L = -\frac{1}{mC_s} \sum_{i=1}^{mC_s} \boldsymbol{y}_s^i \log(P(\phi(\hat{\boldsymbol{x}}_s^i))), \tag{8.5}$$

m is the number of anchors in each class.

It is the expected that the proposed model can handle both multiple source domains with homogeneous and heterogeneous label spaces, and by minimizing equation (8.4), we can guarantee data distribution on a global level. However, when multi-source domains have heterogeneous label spaces, for the second condition, introducing unshared anchors to a source domain can have a negative influence. To solve this problem, a source local category discriminator is designed to ensure a source domain to optimize its own class anchors on the category level while reducing the influence of unshared source anchors. The source category discriminator can be expressed as:

$$\begin{aligned} L_{bce} =& \sum_{c=1}^{C_s} L(P_c(\phi(\boldsymbol{x}_{s_k})), I(\boldsymbol{y}_{s_k}, c)), \\ k =& 1, \cdots, K. \end{aligned} \tag{8.6}$$

where $I(\boldsymbol{y}_{s_k}, c) = 1_{y_{s_k}=c}$. For the kth source domain with label space $\mathcal{C}_{s_k}$, the category discriminators $\{P_c\}_{c \in \mathcal{C}_{s_k}}$ are optimized. The source data generator is further controlled by:

$$G = \arg\min_{G} \; L(P_c(\phi(G(\boldsymbol{y}_s))), I(\boldsymbol{y}_s, c)), \tag{8.7}$$

The loss function of the source data generator satisfying the first condition is further updated as the combination of global-level and category-level constraints:

$$\begin{aligned} L_G =& L(P(\phi(G(\boldsymbol{y}_s))), \boldsymbol{y}_s) + \\ & L(P_c(\phi(G(\boldsymbol{y}_s))), I(\boldsymbol{y}_s, c)). \end{aligned} \tag{8.8}$$

For the second condition, denote the source class center as the mean value of the generated source samples from the corresponding category, the source generated center is:

$$\begin{aligned} \hat{\boldsymbol{v}}_s^c =& \frac{1}{m} \sum_{i=1}^{m} \phi(G(\boldsymbol{y}_s^{ci})), \\ c =& 1, \cdots, C_s. \end{aligned} \tag{8.9}$$

In every source domain, we minimize the distance between source samples and its own class centers, which is expressed as:

$$\begin{aligned} L_c =& \sum_{c=1}^{C_{s_k}} \left\| \sum_{i=1}^{b} P_c(\phi(\boldsymbol{x}_{s_k}^i))\phi(\boldsymbol{x}_{s_k}^i) - \hat{\boldsymbol{v}}_s^c \right\|^2, \\ k =& 1, \cdots, K. \end{aligned} \tag{8.10}$$

b is the batch size.

To enhance the representation of the generated source class anchors, except for forcing source samples to be close the corresponding source anchors, we enlarge the distance between anchors from different classes to learn clear boundaries among classes. Denote a positive sample of a source anchor as $\hat{\boldsymbol{x}}_s^+$ and a negative sample as $\hat{\boldsymbol{x}}_s^-$, contrastive loss (Khosla et al., 2020) is employed to separate anchors from different classes, which is:

$$L_{con} = -log\frac{exp(\psi(\hat{\boldsymbol{x}}_s, \hat{\boldsymbol{x}}_s^+)/\tau)}{exp(\psi(\hat{\boldsymbol{x}}_s, \hat{\boldsymbol{x}}_s^+)/\tau) + \sum_{i=1}^{C_s-1} exp(\psi(\hat{\boldsymbol{x}}_s, \hat{\boldsymbol{x}}_s^{i-})/\tau)} \tag{8.11}$$

where ψ is a distance measurement calculating the similarity between samples and τ is the temperature factor.

The total loss of optimizing the source data generator is then expressed as:

$$L_{gen} = L_G + L_c + L_{con}. \tag{8.12}$$

8.3.3 Target Model Adaptation

In the domain adaptation procedure, source data access is unavailable. To perform the source model on the target domain, we adopt cluster matching to group the target samples to the source categories which can classify known samples. At the same time, we learn thresholds to identify both source private classes and unknown target samples to reduce the influence of unshared categories. To achieve this, target samples are pseudo-labeled first to provide self-supervision of extracting invariant information and select known target samples. To collect highly confident target pseudo labels, both the predictions of clustering and classification are considered to reduce the pseudo label noise.

After feeding the target samples to the pre-trained unified learning model and category discriminators, the predicted outputs are:

$$\boldsymbol{\omega}_G = [\omega_G^1, \cdots, \omega_G^{C_s}] = P(\phi(\boldsymbol{x}_t)), \tag{8.13}$$

and

$$\boldsymbol{\omega}_S = [\omega_S^1, \cdots, \omega_S^{C_s}] = [P_c(\phi(\boldsymbol{x}_t)), \cdots, P_{C_s}(\phi(\boldsymbol{x}_t))]. \tag{8.14}$$

where $\boldsymbol{\omega}_P$ and $\boldsymbol{\omega}_{P_c}$ denote the probability vectors indicating the degrees of a target sample belonging to the source classes. The predicted target label is:

$$\begin{aligned} \hat{\boldsymbol{y}}_t =& \underset{c}{\text{Max}} \frac{\boldsymbol{\omega}_G + \boldsymbol{\omega}_S}{2}, \\ \boldsymbol{\omega}_G,&\boldsymbol{\omega}_S \in \mathbb{R}^{C_s}, c \in \mathcal{C}_s. \end{aligned} \tag{8.15}$$

Since there are unknown classes in the target domain and unshared classes in the source domain, if all pseudo target labels are used to calculate target clustering centers, unrelated information from unshared classes can degrade the classification performance and result in negative transfer when adapting the source model to the target domain. To avoid this, we first learn a threshold that divides the known and unknown samples from the target domain, which is:

$$a_o = \sum_{c=1}^{C_s} \frac{(\boldsymbol{\omega}_G + \boldsymbol{\omega}_S) \log(P(\phi(\boldsymbol{x}_t)))}{2 \log(C_s)}. \tag{8.16}$$

If the maximum probability value of a target sample is higher than a_o, we regard it as a sample from the known classes. Otherwise, we regard it as a sample from the unknown classes with label $C_s + 1$, and these unknown samples are not used to calculate clustering centers.

After removing the unknown samples defined by the learned threshold a_o, we gather target samples with label space $\mathcal{C}_f$. This sample set can include target unknown samples which are given source private labels. To identify these target samples, we first adopt another threshold to select target samples with highly confident pseudo labels. Denote the pseudo label of cth class as $\hat{\boldsymbol{y}}_t^c$, the corresponding maximum probability returned by source classifiers as ω_{max}^c, the threshold to select highly confident target labels is:

$$a_p^c = \text{med}\{\omega_{max}^{ci}\}_{i=1}^{n_c}, \tag{8.17}$$

where n_c is the number of target samples divided into the cth source class. We group samples whose maximum probability is larger than a_p^c as confident target samples.

Initial target clustering centers are calculated based on these selected target samples, which is:

$$\boldsymbol{v}_t^c = \frac{\sum_{i=1}^{\hat{n}_c}(\boldsymbol{\omega}_G^i + \boldsymbol{\omega}_S^i) \cdot \phi(\boldsymbol{x}_t^i)}{\sum_{i=1}^{\hat{n}_c}(\boldsymbol{\omega}_G^i + \boldsymbol{\omega}_S^i)}. \tag{8.18}$$

To further reduce the target pseudo label noise by removing the source private classes, a strategy is designed to identify source unshared classes. First, when feeding the target samples to the source classifier, if there is no target sample classified into a source class, we remove the corresponding source class as a source private class. This operation can be expressed as:

$$\begin{aligned} \boldsymbol{N} &= \mathrm{Count}(\{\omega_{max}^c\}), \\ \boldsymbol{N} &= [n_1, \cdots, n_{C_s}], c \in \mathcal{C}_s. \end{aligned} \tag{8.19}$$

Count means the operation to count the number of target samples with label c. If $n^c > 0$, corresponding class c is added to the common label set of source and target domains, otherwise we remove the corresponding class as a source private class. Furthermore, assuming the cluster centers of the common categories from the source and target domains are close to each other, we calculate the similarity between target clustering centers and source anchors, which is:

$$r = \arg\min_i \mathrm{Dis}(\boldsymbol{v}_t^i, \hat{\boldsymbol{v}}_s^c). \tag{8.20}$$

where Dis is cosine similarity, r is the class index of the target clustering center which is closest to the cth source class center. If the cth target cluster center gets the closest source class anchor as $\hat{\boldsymbol{v}}_s^c$, where $r = c$, we regard that the cth class is a common category of source and target domains. Denote the final common label set selected by equations (8.19) and (8.20) as $\mathcal{C}$, the target pseudo label is initialized as:

$$\tilde{\boldsymbol{y}}_t = \arg\min_c \ \mathrm{Dis}(\phi(\boldsymbol{x}_t), \boldsymbol{v}_t), \boldsymbol{v}_t = \{\boldsymbol{v}_t^c\}_{c \in \mathcal{C}}, \tag{8.21}$$

Combining the predictions returned by equation (8.15), we design a memory bank to store the samples whose predictions $\tilde{\boldsymbol{y}}_t$ and $\hat{\boldsymbol{y}}_t$ are the same, which is denoted as $\mathcal{D}'_t = \{\boldsymbol{x}_t\}_{\tilde{\boldsymbol{y}}_t = \hat{\boldsymbol{y}}_t}$. The target cluster centers and pseudo labels are then updated as:

$$\begin{aligned}
\boldsymbol{v}_t^c &= \frac{\sum_{i=1}^{\hat{n}'_c} \mathbb{1}_{\tilde{\boldsymbol{y}}_t^i = c} \cdot \phi(\boldsymbol{x}_t^i)}{\sum_{i=1}^{\hat{n}'_c} \mathbb{1}_{\tilde{\boldsymbol{y}}_t^i = c}}, \\
\tilde{\boldsymbol{y}}_t &= \arg\min_c \, \mathrm{Dis}(\phi(\boldsymbol{x}_t), \boldsymbol{v}_t), \\
\boldsymbol{v}_t &= \{\boldsymbol{v}_t^c\}_{c \in \mathcal{C}},
\end{aligned} \tag{8.22}$$

$\hat{n}'_c$ is the number of samples in the cth class stored in the memory bank.

Employing the pseudo labels obtained by equation (8.22), freeze the classifier layer, the source model is adapted to the target domain using a self-supervision strategy by fine-tuning the feature extractor, which is:

$$\phi = \arg\min_{\substack{\phi \\ x_t \sim \mathcal{D}'_t}} L_\phi(P(\phi(\boldsymbol{x}_t))), \tilde{\boldsymbol{y}}_t), \tag{8.23}$$

where

$$L_\phi = -\frac{1}{\sum_{c=1}^{C} \hat{n}'_c} \sum_{i=1}^{\sum_{c=1}^{C} \hat{n}'_c} \tilde{\boldsymbol{y}}_t \log(P(\phi(\boldsymbol{x}_t^i))). \tag{8.24}$$

Following previous data-free domain adaptation methods, information maximization loss is employed to balance the domain (Liang et al., 2020; Hu et al., 2017):

$$L_{div} = \sum \bar{\boldsymbol{p}}_t \log(\bar{\boldsymbol{p}}_t), \tag{8.25}$$

where

$$\bar{\boldsymbol{p}}_t = \frac{1}{\sum_{c=1}^{C} \hat{n}'_c} \sum_{i=1}^{\sum_{c=1}^{C} \hat{n}'_c} P(\phi(\boldsymbol{x}_t^i)). \tag{8.26}$$

Except for the label-level constraint controlled by self-supervision, to better transform the target data distribution to the source feature space, data-level constraints are adopted to group samples from the same classes, and enlarge the distance among known target classes as well as between known and unknown samples. To match the target samples to the shared source classes, the distance between target samples and the source anchors is minimized by:

$$\begin{aligned} L_{st} &= \left\| \sum_{i=1}^{b} P(\phi(\boldsymbol{x}_t^i))\phi(\boldsymbol{x}_t^i) - \hat{\boldsymbol{v}}_s \right\|^2, \\ \hat{\boldsymbol{v}}_s &= \{\hat{\boldsymbol{v}}_s^c\}_{c\in\mathcal{C}} \end{aligned} \tag{8.27}$$

To separate known classes from each other, contrastive loss is adopted on the pseudo-labeled data, which is:

$$L_{cont} = -log \sum_{c\in\mathcal{C}} \frac{exp(\psi(\boldsymbol{x}_t, \boldsymbol{x}_t^{c+})/\tau)}{exp(\psi(\boldsymbol{x}_t, \boldsymbol{x}_t^{+})/\tau) + \sum_{i\neq c} exp(\psi(\boldsymbol{x}_t, \boldsymbol{x}_t^{i-})/\tau)} \tag{8.28}$$

where $\boldsymbol{x}_t$ is any target sample belonging to the cth class, $\boldsymbol{x}_t^+$ is the positive sample from the same class, $\boldsymbol{x}_t^-$ indicates a negative sample from the other classes, ψ is the similarity measurement which is the same as that in equation (8.11), and τ is the temperature factor.

To separate the known and unknown target samples, we enlarge the distance between the samples in memory bank $\mathcal{D}_t'$ and the unknown samples defined by equation (8.16). As the size of the unknown samples is often larger than the known samples, we rank the unknown samples by their entropy assumptions and select $\frac{1}{3}$ of the samples with the highest entropy loss, denoted as $\mathcal{D}_t^u = \{\boldsymbol{x}_t^{ui}\}_{i=1}^{n^u}$. We maximize the distance between the known and unknown samples by:

$$\begin{aligned} L_{uk} =& \arg\max_{\phi} \\ =& \left\| \frac{1}{\sum_{c=1}^{C} \hat{n}_c'} \sum_{i=1}^{\sum_{c=1}^{C} \hat{n}_c'} h(\phi(\boldsymbol{x}_t^i)) - \frac{1}{n^u} \sum_{j=1}^{n^u} h(\phi(\boldsymbol{x}_t^{uj})) \right\|_{\mathcal{H}}^2, \end{aligned} \tag{8.29}$$

The total loss function of adapting source model to target domain is:

$$L_{total} = L_\phi + \alpha L_{div} + \beta L_{st} + \gamma L_{cont} + \lambda L_{uk}. \tag{8.30}$$

The processing of the proposed unified learning model for multi-source-free domain adaptation is described in Algorithms 13 and 14.

Algorithm 13 ULMSFDA: Source model training.

```
Input: Source domains;
for $\epsilon = 1,\ \epsilon < \mathcal{I}_s,\ \epsilon + +,$ do
    Update classifier by unifying source knowledge as in equation (8.1);
    Update source category discriminator as in equation (8.6);
    Calculate entropy-loss of source data generator as in equation and (8.8);
    Calculate source generated class center as in equation and (8.9);
    Calculate contrastive loss of source data generator as in equations (8.10) and (8.11);
    Update source data generator as in equation (8.12);
end for
Output: Unified learning model, source category discriminator, source generated center.
```

Algorithm 14 ULMSFDA: Target model adaptation.

```
Input: Unified learning model, source category discriminator, source generated center, target domain;
for $\epsilon = 1$, $\epsilon < \mathcal{I}_t$, $\epsilon + +$, do
    Calculate target pseudo labels as in equation (8.15);
    Learn the threshold to identify unknown samples as in equation (8.16);
    Select confident target labels as in equation (8.17);
    Calculate target class centers as in equation (8.18);
    Identify common classes as in equation (8.20);
    Predict target labels using clustering as in equation (8.21);
    Update target class centers and pseudo labels as in equation (8.22);
    Calculate loss of self-supervision as in equations (8.24);
    Calculate information maximization loss as in equations (8.25);
    Adapt target data to source generated centers as equation (8.27);
    Calculate contrastive loss on target domain as equation (8.28);
    Enlarge distance of known and unknown classes as equation (8.29);
    Fine-tune feature extractor as equation (8.30);
end for
Output: Target labels.
```

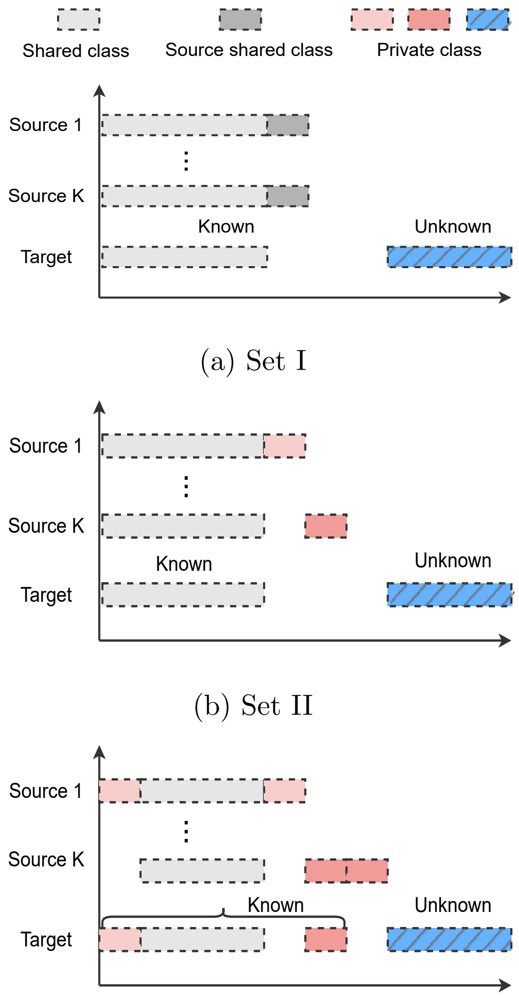
8.4 Experiments

In this section, the proposed unified learning model is validated on three popular real-world visual datasets under three settings. The datasets include Office31, OfficeHome and DomainNet. Office31 camprises 31 object categories collected from three domains. Three tasks can be built: $A, W \rightarrow D$; $A, D \rightarrow W$; $W, D \rightarrow A$. OfficeHome consists of four domains with each domain containing 65 categories, the tasks are $A, C, P \rightarrow R$; $A, C, R \rightarrow P$; $A, P, R \rightarrow C$; $C, P, R \rightarrow A$. DomainNet consists of 6 domains with each domain consisting of 345 categories. Following previous studies (Li et al., 2021a; Saito and Saenko, 2021), we run subset experiments which are $R, S \rightarrow P$; $P, S \rightarrow R$; $P, R \rightarrow S$.

All the experiments are classification tasks under the multi-source domain adaptation scenario, where both source domains with homogeneous and heterogeneous label spaces are applied to validate the proposed method. Harmonic mean (HM) on the accuracy of known and unknown classes is employed to measure the performance of the proposed unified learning model (Fu et al., 2020). The results are the mean values of three repeat runs on each task. The experiment settings are shown in Fig. 8.3. For multi-source domains with homogeneous label space (Set I), we follow the work in (You et al., 2019; Li et al., 2021a) to set the known and unknown classes. For multi-source domains with heterogeneous label spaces (Sets II and III), the label set of the source and target domains are shown in Table 8.2.

The compared baselines include heterogeneous single source and multi-source domain adaptation methods with and without source data, all results are collected under universal setting. Methods with source access include:

- RTN: Residual transfer networks (Long et al., 2016);
- IWAN: Importance weighted adversarial nets (Zhang et al., 2018);



(a) Set I

(b) Set II

(c) Set III

Figure 8.3 : Settings of universal multi-source-free domain adaptation.

Table 8.2 : Label set division under three settings. In source domain, the division is listed as source shared/private classes. In target domain, the division is listed as known/unknown classes.

Dataset	Domain	Set II	Set II	Set III
Office31	S1	20	10/5	6/7
	S2	20	10/5	6/7
	T	10/11	10/11	10/11
OfficeHome	S1	15	10/5	4/7
	S2	15	10/5	4/7
	S3	15	10/5	4/7
	T	10/50	10/40	10/40
DomainNet	S1	200		
	S2	200	-	-
	T	150/145		

- PADA: Partial adversarial domain adaptation (Cao et al., 2018b);
- ATI: Open set domain adaptation for image and action recognition (Busto et al., 2018);
- OSBP: Open set domain adaptation by backpropagation (Saito et al., 2018);
- UAN: Universal domain adaptation (You et al., 2019);
- CMU: Learning to detect open classes for universal domain adaptation (Fu et al., 2020);
- DCC: Domain consensus clustering for universal domain adaptation (Li et al., 2021a);
- OVA: One vs all net (Saito and Saenko, 2021);

Source-free methods include:

- SHOT: Source hypothesis transfer with information maximization (Liang et al., 2020);
- USFDA: Universal source-free domain adaptation (Kundu et al., 2020a);
- UMAD: Universal model adaptation under domain and category shift (Liang et al., 2021b);
- OneRing: One ring (Yang et al., 2022);
- UB2DA: Universal black-box domain adaptation (Deng et al., 2021).

Results of the partial (PADA) and open-set (ATI, OSBP) domain adaptation baselines are re-run under universal settings. All the compared results are obtained from previous publications. For single source-free domain adaptation methods, we take the average predictions from all source domains as the multi-source results similar to previous studies.

$ResNet50$ is employed as the backbone on datasets Office31 and OfficeHome complemented by PyTorch, while $ResNet101$ is applied on DomainNet. Parameters are updated based on backpropagation with stochastic gradient descent, the momentum is 0.9, the learning rate η follows the same strategy in (Ganin and Lempitsky, 2015), which is $\eta = \frac{\eta_0}{(1+10\epsilon)^{0.75}}$, where $\eta_0 = 0.01$, ϵ is the training progress which changes linearly from 0 to 1. The learning rate of the shared network is one tenth of the other layers. Batch size $n_b = 64$, and the smoothing parameter $\mu = 0.1$.

8.4.1 Results and Analysis

Tables 8.3, 8.4 and 8.5 show the HM on tasks from DomainNet, OfficeHome and Office31 respectively. The proposed unified learning model (ULMSFDA) achieves the highest average performance on most tasks and datasets. On the datasets DomainNet and OfficeHome, the proposed ULMSFDA performs better than both baselines with and without source data. The averaged HM is improved by 0.7% and 3.4% respectively. On tasks $R, S \to P$ and $P, S \to R$ from DomainNet, the method OneRing shows higher HM than the proposed ULMSFDA. OneRing regards the non-ground-truth category as an unknown category to train a classifier whose last dimension indicates the probability of the sample is unknown, which requires extra parameter leaning in the model to distinguish unknown samples from known. The proposed ULMSFDA learns a threshold to identify unknown samples, which is easy to calculate without introducing extra parameters. The performance of the proposed ULMSFDA on tasks from the datasets OfficeHome and Office31 is better than OneRing. Furthermore, compared with OneRing and other baselines, the proposed method can handle more complex settings for universal domain adaptation. On the dataset Office31, our method achieves the second best average performance with a very small gap between it and and the best method UMAD. UMAD designs an informative consistency score to measure the similarity between samples and detect unknown classes. To learn the score, it generates negative samples by mixing any two known target samples. This mix can introduce more generated samples than the proposed ULMSFDA. For datasets containing a large numbers of samples and categories, where there are more unknown samples than known samples, such as OfficeHome and DomainNet, the approach of generating negative samples is not as significant as when it is used on a samll dataset. The proposed ULMSFDA has an advantage on large datasets over UMAD which introduces too many negative samples.

As claimed in previous studies (Fu et al., 2020; Yang et al., 2022), classification accuracy based on per-class is not a reasonable evaluation of universal domain adaptation, because high accuracy over a per-known-class can lead to a high mean result on both known and unknown classes even when the accuracy of unknown classes is 0. However, it is expected that universal domain adaptation methods should guarantee accuracy on both known and unknown classes. Thus, we take HM as the evaluation metric to measure the performance of the proposed method. Considering many existing universal domain adaptation methods only provide classification accuracy results, we also compare classification accuracy on the datasets OfficeHome and Office31 under Set I to provide sufficient validation of the proposed ULMSFDA. Tables 8.6 and 8.7 show the classification accuracy of the proposed method and the baselines. It can be seen that the proposed ULMSFDA outperforms the existing methods on both datasets.

Tables 8.8 and 8.9 show the HM of the proposed methods and baselines ResNet and SHOT under Set II, where multi-source domains have homogeneous label spaces. Tables 8.10 and 8.11 show their results under Set III, where multi-source domains have heterogeneous label spaces. It can be seen that the proposed ULMSFDA

Table 8.3 : HM (%) on dataset DomainNet of the ULMSFDA and baselines under Set I

Method	SF	R,S→P	P,S→R	P,R→S	Avg
RTN	×	29.5	32.1	28.7	30.1
IWAN	×	32.1	35.2	31.1	32.8
PADA	×	26.7	27.4	27.3	27.1
ATI	×	29.8	32.4	28.9	30.4
OSBP	×	31.8	33.6	30.6	32.0
UAN	×	41.3	42.8	38.9	41.0
CMU	×	48.5	50.9	45.4	48.3
DCC	×	47.6	56.5	43.4	49.2
OVA	×	48.9	56.3	44.4	49.8
SHOT	✓	34.6	33.6	29.6	32.6
UMAD	✓	41.1	57.2	43.2	47.1
OneRing	✓	**50.8**	**57.9**	45.3	51.3
UB2DA	✓	48.1	54.3	45.6	49.3
ULMSFDA	✓	50.3	57.5	**48.1**	**52.0**

Table 8.4 : HM (%) on dataset OfficeHome of the ULMSFDA and baselines under Set I

Method	SF	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
RTN	×	45.6	44.4	38.3	43.3	42.9
IWAN	×	47.6	46.2	41.5	45.7	45.3
PADA	×	44.1	42.3	34.2	40.2	40.2
ATI	×	46.6	45.2	41.0	44.6	44.4
OSBP	×	46.2	45.7	40.6	45.3	44.5
UAN	×	59.2	58.2	50.6	58.3	56.6
CMU	×	64.5	63.6	55.0	63.3	61.6
DCC	×	70.1	68.4	68.9	73.2	70.2
OVA	×	79.1	74.9	59.5	71.3	71.2
SHOT	✓	41.0	31.0	33.9	56.7	40.7
UMAD	✓	78.2	73.7	59.1	69.4	70.1
OneRing	✓	78.8	72.1	62.7	73.4	71.8
UB2DA	✓	76.3	70.0	61.1	74.3	70.4
ULMSFDA	✓	**81.7**	**76.5**	**65.6**	**76.8**	**75.2**

Table 8.5 : HM (%) on dataset Office31 of the ULMSFDA and baselines under Set I

Method	SF	A,W→D	A,D→W	W,D→A	Avg
RTN	×	52.7	52.4	48.5	51.2
IWAN	×	53.0	52.1	49.7	51.6
PADA	×	52.8	51.1	46.0	50.0
ATI	×	53.0	51.8	48.7	51.2
OSBP	×	54.2	52.9	50.0	52.3
UAN	×	65.6	64.6	60.2	63.5
CMU	×	74.3	73.3	71.8	73.1
DCC	×	88.6	78.9	73.1	80.2
SHOT	✓	79.0	77.8	68.2	75.0
USFDA	✓	83.4	85.2	86.0	84.9
UMAD	✓	**92.2**	89.7	**87.5**	**89.8**
OneRing	✓	90.9	89.5	85.2	88.5
UB2DA	✓	84.4	85.4	91.0	86.9
ULMSFDA	✓	90.3	**93.7**	84.8	89.6

Table 8.6 : Accuracy (%) on dataset OfficeHome of the ULMSFDA and baselines under Set I

Method	SF	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
RTN	×	86.0	77.0	60.0	68.7	72.9
IWAN	×	85.6	77.1	56.5	74.3	73.4
PADA	×	77.8	71.6	40.0	62.3	62.9
ATI	×	85.3	77.1	57.0	73.8	73.3
OSBP	×	76.8	65.9	49.1	63.8	63.9
UAN	×	86.7	80.3	61.7	**79.5**	77.1
SHOT	✓	70.1	68.4	**68.9**	73.2	70.2
USFDA	✓	87.6	81.3	62.2	77.1	77.1
UB2DA	✓	92.9	84.2	57.3	76.8	77.7
ULMSFDA	✓	**90.2**	**85.4**	59.5	76.3	**77.9**

Table 8.7 : Accuracy (%) on dataset Office31 of the ULMSFDA and baselines under Set I

Method	SF	A,W→D	A,D→W	W,D→A	Avg
ResNet	×	85.7	82.8	80.1	82.9
IWAN	×	87.1	87.6	85.2	86.6
PADA	×	86.3	82.3	69.0	79.2
ATI	×	87.2	86.0	80.2	84.5
OSBP	×	79.3	69.9	53.9	67.7
UAN	×	92.3	90.2	85.3	89.2
SHOT	✓	88.6	78.9	73.1	80.2
USFDA	✓	93.1	90.4	87.1	90.2
OneRing	✓	92.1	86.8	81.5	86.8
UB2DA	✓	93.3	90.6	**91.2**	91.7
ULMSFDA	✓	**95.3**	**94.7**	86.5	**92.2**

performs better than the other methods. Set II and Set III are situations rarely explored in previous universal domain adaptation methods. One advantage of the proposed method is that it can handle both homogeneous and heterogeneous source label spaces without training an independent model in each source domain. Many previous universal domain adaptation methods cannot deal with multiple source domains simultaneously. For a target task, if there are multiple source domains, they have to adapt each pair of source and target domains to predict the target task. The proposed ULMSFDA learns one model to predict multiple tasks, it is flexible enough to combine knowledge from multiple source domains to explore more information to complete the target task.

Table 8.8 : HM (%) on dataset OfficeHome of the ULMSFDA and baselines under Set II

Method	SF	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
ResNet	✓	**82.1**	**75.3**	54.8	68.0	70.1
SHOT	✓	77.3	70.2	56.4	72.1	69.0
ULMSFDA	✓	78.8	72.8	**62.4**	**76.8**	**72.7**

Table 8.9 : HM (%) on dataset Office31 of the ULMSFDA and baselines under Set II

Method	SF	A,W→D	A,D→W	W,D→A	Avg
ResNet	✓	90.7	85.2	62.5	79.5
SHOT	✓	88.6	90.0	79.9	89.2
ULMSFDA	✓	**93.3**	**92.7**	**85.5**	**90.5**

Table 8.10 : HM (%) on dataset OfficeHome of the ULMSFDA and baselines under Set III

Method	SF	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
ResNet	✓	**78.5**	**71.4**	51.8	64.9	66.7
SHOT	✓	73.8	66.9	54.3	71.0	66.5
ULMSFDA	✓	73.1	70.9	**59.2**	**73.2**	**69.1**

Table 8.11 : HM (%) on dataset Office31 of the ULMSFDA and baselines under Set III

Method	SF	A,W→D	A,D→W	W,D→A	Avg
ResNet	✓	82.7	71.7	54.3	69.6
SHOT	✓	76.7	77.7	71.6	75.3
ULMSFDA	✓	**95.4**	**90.7**	**81.3**	**89.1**

8.4.2 Ablation Study

Tables 8.13, 8.14 and 8.15 show the results of the ablation study on the dataset OfficeHome under set I, II and III respectively. We evaluate three modules in the domain adaptation procedure when training the target model: the influence of matching target data to generated source data is reflected by loss function L_{st}, the influence of contrastive learning is reflected by loss function L_{cont}, the influence of separating known and unknown classes is reflected by loss function L_{uk}. The details of ablation study setting are listed in Table 8.12.

Table 8.12 : Setting of ablation study

Method	generated data matching	Contrastive loss	Known and unknown separation
L_{st}	×	✓	✓
L_{cont}	✓	×	✓
L_{uk}	✓	✓	×
Proposed	✓	✓	✓

It can seen that the model trained without contrastive loss L_{cont} performs worst under the three settings, which indicates that the contrastive loss which forces samples from the same class to be close to each other and separates samples from different classes is the most important module for the proposed method. The performance of the model without loss function L_{uk} shows a larger decrease in most settings. This indicates that the operation to enlarge the distance between the known and unknown classes also plays an essential role in guaranteeing the transfer performance. The employment of generated matching can have a positive influence on the proposed ULMSFDA. Without adapting target data to source generated centers controlled by

L_{st}, the value of HM reduced under all settings.

Table 8.13 : Ablation study (HM (%)) on dataset OfficeHome under Set I

Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
L_{st}	82.2	75.4	64.4	**76.9**	74.7
L_{cont}	78.4	72.1	62.3	74.8	71.9
L_{uk}	81.7	75.5	61.7	76.3	73.8
Proposed	**81.7**	**76.5**	**65.6**	76.8	**75.2**

Table 8.14 : Ablation study (HM (%)) on dataset OfficeHome under Set II

Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
L_{st}	78.1	71.9	60.9	76.3	71.8
L_{cont}	72.8	68.6	61.2	75.1	69.4
L_{uk}	78.5	72.6	**62.9**	76.3	72.6
Proposed	**78.8**	**72.8**	62.4	**76.8**	**72.7**

8.4.3 Influence of Source Category Discriminator

In this section, we validate the influence of the source pre-trained model with and without the source category discriminator. The source category discriminator optimized by loss function L_{bce} is designed to learn high representative generated centers when multiple source domains have heterogeneous label spaces. We remove this module in three settings respectively during the source unified learning model training and transfer the source model without the source category discriminator to the target domain to test its performance. The results are shown in Table 8.16, and

Table 8.15 : Ablation study (HM (%)) on dataset OfficeHome under Set III

Method	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
L_{st}	**73.4**	**70.9**	**59.2**	71.9	68.9
L_{cont}	68.5	64.4	54.3	69.3	64.1
L_{uk}	72.7	70.7	57.9	71.9	68.3
Proposed	73.1	**70.9**	**59.2**	**73.2**	**69.1**

the method L_{bce} indicates the model is trained without source category discriminator. It can be seen that the source category discriminator has a positive influence on the proposed method even under the setting where source domains have the same label spaces. The source category discriminator can learn specific source information especially that which is contained in source private classes, and it also has the advantage of combining the knowledge of shared source categories. The specific information can help us identify source private classes when pseudo-labeling the target samples, while the common information extracted by unifying knowledge from shared categories can enrich the transfer information to help predict the target known classes.

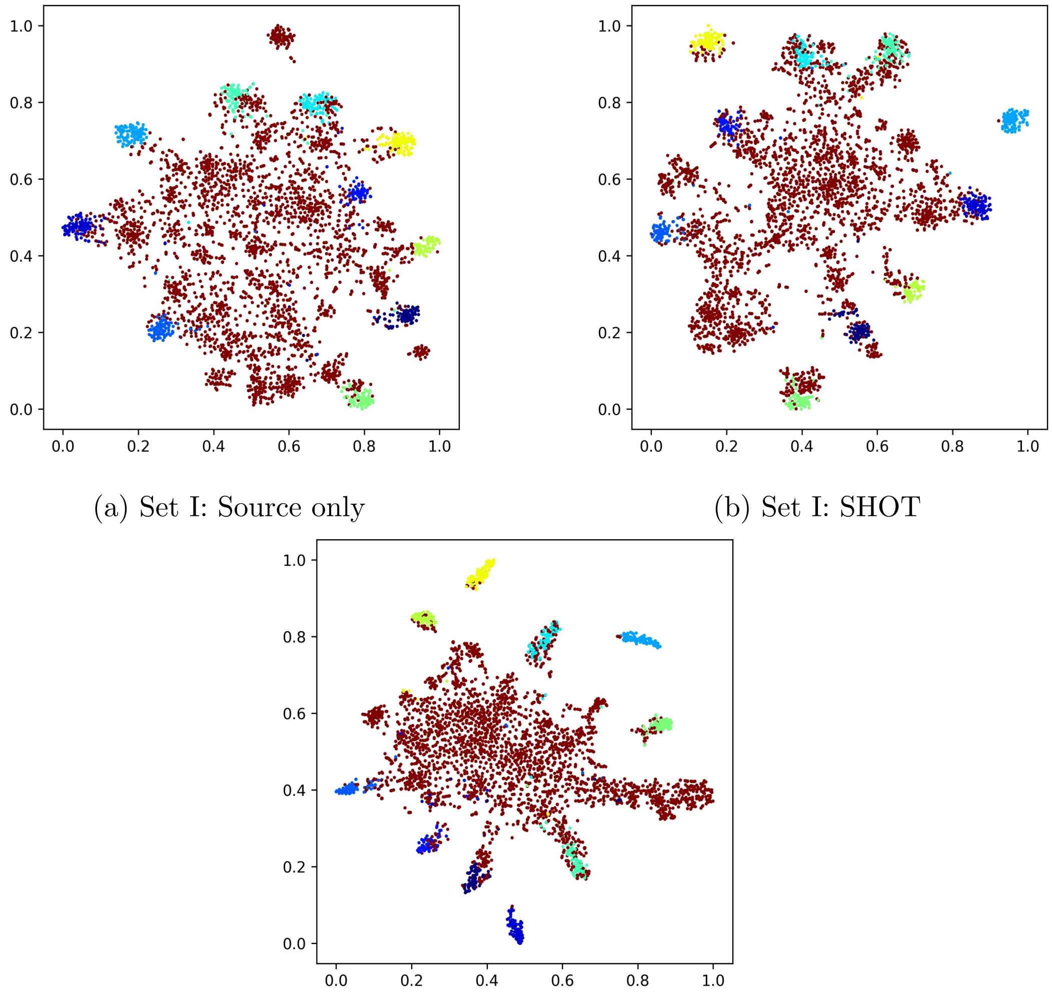
8.4.4 Visualization Analysis

This section provides a visualization analysis of the proposed method under three settings. Taking task $A, C, P \rightarrow R$ from the dataset OfficeHome as an example, Figs. 8.4, 8.5 and 8.6 show the T-SNE visualization (Maaten and Hinton, 2008) of the proposed method and baseline SHOT. "Source only" refers to the model without domain adaptation based on $ResNet50$. It can be seen that the proposed method can divide target samples from known classes with clear decision boundaries. For unknown classes, compared with the source-only model without transfer in which

Table 8.16 : HM (%) on dataset OfficeHome with and without source category discriminator

Method	Setting	A,C,P→R	A,C,R→P	A,P,R→C	C,P,R→A	Avg
Set I	L_{bce}	80.1	73.3	62.4	74.0	72.5
	Proposed	**81.7**	**76.5**	**65.6**	**76.8**	**75.2**
Set II	L_{bce}	75.9	72.1	**62.4**	70.9	70.3
	Proposed	**78.8**	**72.8**	**62.4**	**76.8**	**72.7**
Set III	L_{bce}	**74.1**	67.6	57.2	70.1	67.3
	Proposed	73.1	**70.9**	**59.2**	**73.2**	**69.1**

the known and unknown classes are mixed up, and the baseline SHOT, where too many unknown classes are classified as known classes, when applying the proposed method, the unknown classes are grouped together with a few known samples, which indicates the superiority of the proposed method.

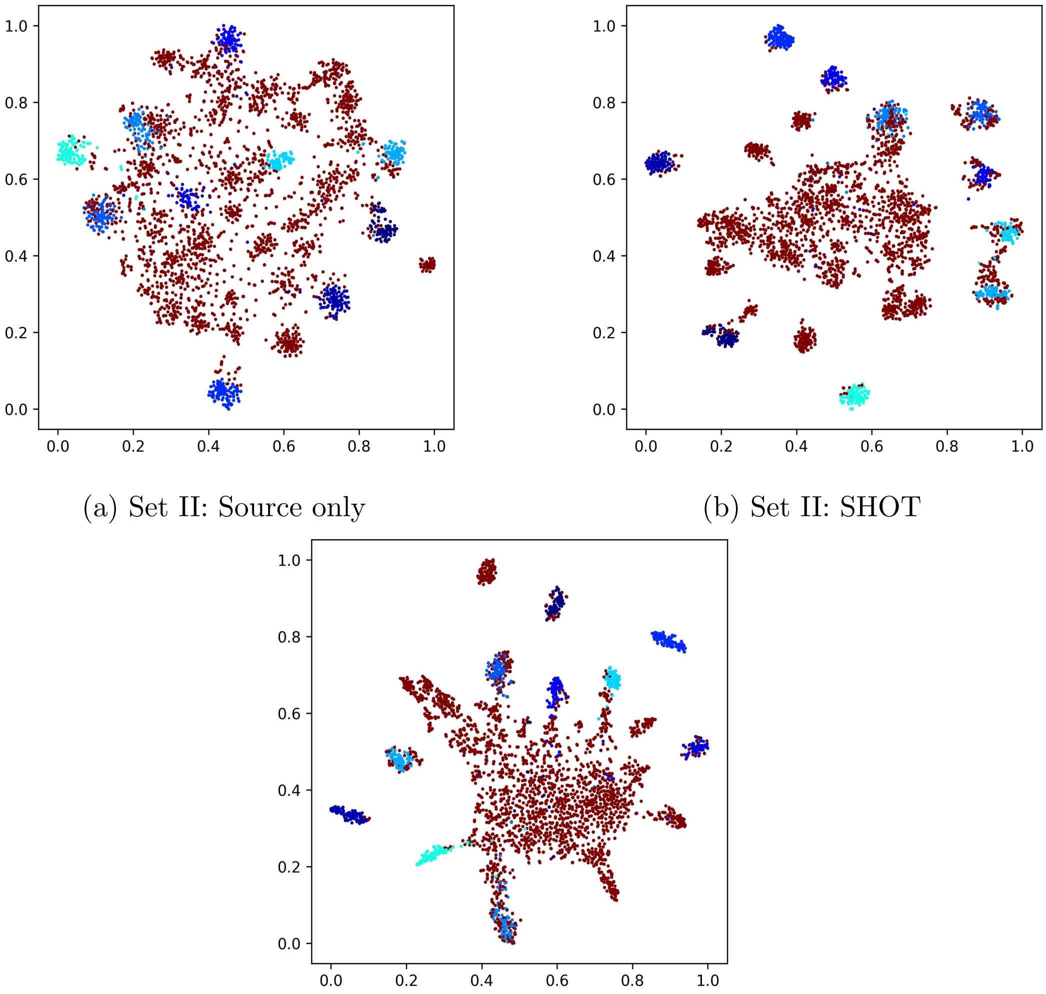


(a) Set I: Source only

(b) Set I: SHOT

(c) Set I: Proposed

Figure 8.4 : T-SNE visualization on target domain RealWorld from dataset Office-Home under Set I.

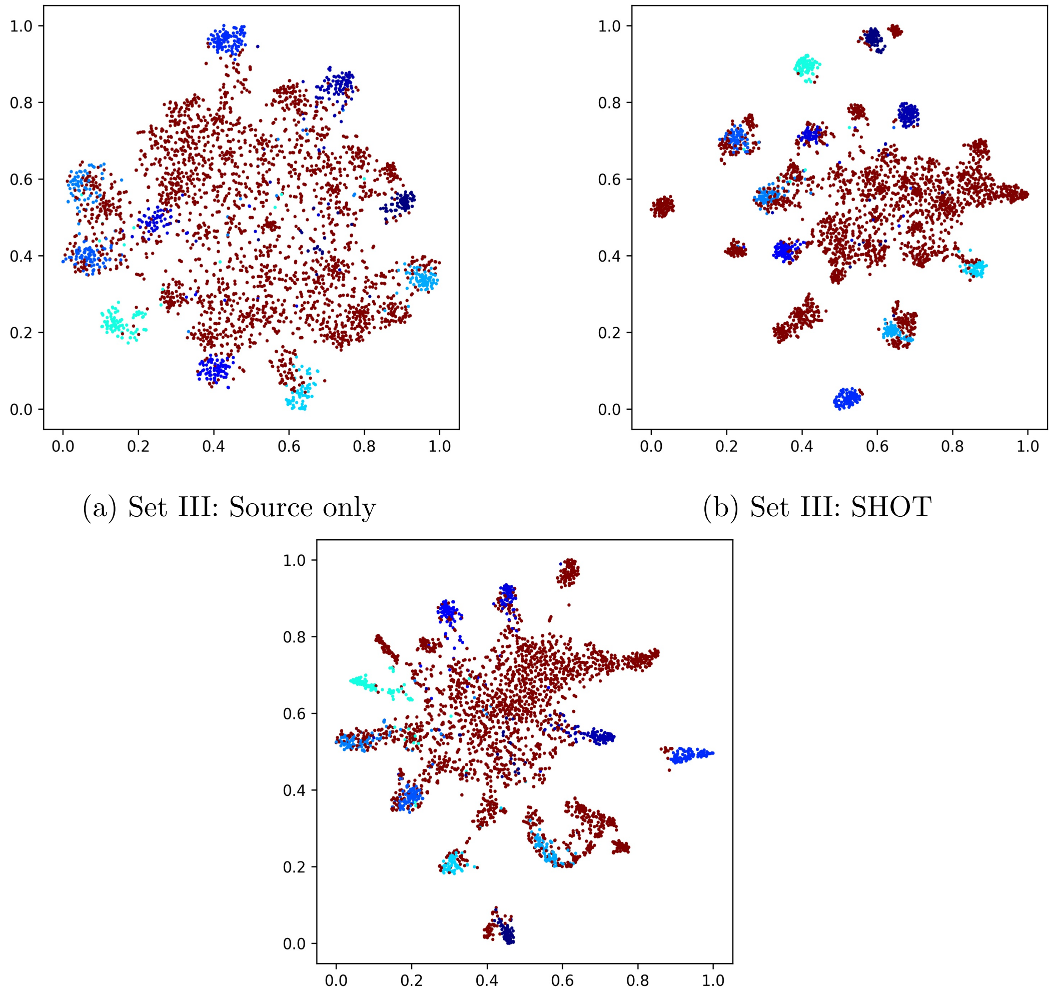


(a) Set II: Source only

(b) Set II: SHOT

(c) Set II: Proposed

Figure 8.5 : T-SNE visualization on target domain RealWorld from dataset Office-Home under Set II.



(a) Set III: Source only

(b) Set III: SHOT

(c) Set III: Proposed

Figure 8.6 : T-SNE visualization on target domain RealWorld from dataset Office-Home under Set III.

8.5 Summary

This chapter proposes a unified learning model for multi-source-free domain adaptation. The unified learning model has the ability to handle both multi-source domains with homogeneous and heterogeneous label spaces without introducing an individual model of each source domain. It pre-trains a unified model to predict known classes shared by source and target domains by combining knowledge from multiple source domains, and learns a source category discriminator to assist in generating high representative source class centers by enhancing the knowledge from the shared classes and reducing the influence of source private classes. Then, the pre-trained source model is applied to the target domain to collect pseudo labels, which are further employed to provide self-supervision of the adapting model and calculate the target clustering centers and threshold to classify known classes and detect unknown samples. The experiments on real-world datasets show that the proposed method outperforms existing universal domain adaptation methods. The HM on the dataset DomainNet is improved by 0.7%, and by 3.4% on dataset OfficeHome. Accuracy is improved by 0.2% and 0.5% on datasets OfficeHome and Office31 respectively.

In the future, we will tackle the sample imbalance problem among classes and source domains. Generally, there are always very large number of unknown samples compared with known samples, so it is easy for the unknown samples to dominate the training of the models which results in the failure of transfer. This problem is worth solving to improve the transfer performance.

Chapter 9

Conclusion and Future Research

9.1 Conclusion

This thesis addresses four questions in transfer learning with multiple source domains. To define what to transfer, a sample and source distillation method is developed to select similar source samples and the dominant source domain. Invariant information extracted from selected source samples is adopted to fit source models to the target domain by matching data distributions on multiple levels. The similarity between source and target domains is defined by constructing a domain discriminator which estimates the degrees of agreement of a target sample belonging to source domains. By ranking the similarities returned by the domain discriminator, dominant source domain is selected to take the prominent place when completing the target task. To compare with the methods without selecting transfer information, the sample and source distillation method achieves higher performance on real-world visual datasets, indicating the superiority of defining transfer information.

To define source contributions, this thesis proposes two methods, including multi-source contribution learning and dynamic classifier alignment. Multi-source contribution learning method is developed to measure the importance of a source domain in completing the target task. Multi-view features are extracted to describe both common and diverse characteristics for source and target domains. Multi-level distribution matching is designed to fill data gap by minimizing the sample distance among the same classes and enlarging that between each pair of different classes. A weight adjustment strategy is built based on pseudo labels predicted by multiple source classifiers to give larger weights to more similar source domains. In addition, a fuzzy rule-based combination is designed to deal with the uncertainty during transfer. Both strategies are expected to reduce negative transfer. Experiments on real-world image classification tasks validate the superiority of measuring source contributions.

Dynamic classifier alignment is developed to learn both importance of multi-view features and contribution of multi-source domains. Auxiliary classifier is generated to learn the importance of multi-view features, which turns the combination of multi-view features with different dimensions to the combination in label space, where the heterogeneity problem can be avoided. Domain discriminator is constructed to measure the contribution of source domains, which uses a pair-wise approach to weigh source combination parameters automatically. Experiments on real-word visual tasks show the superiority of the proposed dynamic classifier alignment method.

To tackle source-free transfer learning, a generally auxiliary model training method is developed to take the advantages of learning from multiple source domains without sharing source data. Source parameters of the local private models are jointly trained by generating a general model to handle richer information. Generation parameters is learned automatically during training by minimizing the cross-entropy loss between the predictions of the generally auxiliary classifier and the ground-truth labels. Class balanced coefficient is adopted to eliminate the influence of minor classes, where the classifier may fail to predict samples occupying a small portion since a large number of samples from other classes dominate the training. Compared with other source-free transfer learning methods, generally auxiliary model outperforms the baselines on real-world visual tasks.

To handle soft information caused by data uncertainty and limited target information, a fuzzy rule-based deep neural network is constructed to adapt source fuzzy rules to the target domain without the access to source data. Source private models under fuzzy rules of each domain are learned by jointly training other source models using an auxiliary learning strategy, where source parameters are shared while source data is preserved. Furthermore, anchor-based alignment is designed to match target samples to the source anchors according to the agreements of clustering a target sample to a source category. Since source data is unavailable, to fit source models better, self-supervised learning based on pseudo labels is employed to train the target feature extractor which transforms target data into a latent feature space close to the source space. To reduce the influence of noisy target labels, a sample selection strategy is designed by combining the predictions of the source model and deep clustering to identify strong target samples, which are then used to update clustering centers that renew pseudo labels with a high level of certainty.

To address transfer learning with heterogeneous label spaces, this thesis explores three scenarios, including partial, open-set and universal transfer learning. In partial transfer learning, shared class samples and private source class samples are divided by ranking entropy assumptions of target samples returned by source classifiers. In open-set transfer learning, known class samples and unknown class samples are identified by defining a threshold in view of the predicted probability values of a target sample belonging to the source classes. In universal transfer learning, both source private and target unknown categories are detected during training to reduce the pseudo label noise to self-supervise the model adaptation without accessing to the source data. The proposed unified learning model is flexible to multi-source domains with homogeneous and heterogeneous label spaces. Experiments on real-world datasets show the superiority of the proposed methods.

9.2 Future Research

There are still some questions in multi-source transfer learning to be solved. First, existing methods we proposed focus on transferring knowledge across domains containing the same type of data. In real-world applications, source and target domains containing different data types are more regular, especially when the granular information is required to detect items in detail. For instance, there are images of animals to be labeled from target domain(s), taking class bird as example, it includes parrot and owl. However, from source domain, only the label bird can be learned, the granular information recognizing parrot and owl which can be referred is textual description from books. What we need to learn is how to transfer textual knowledge to image classification. Zero-shot learning is worthy of exploring in transfer learning to tackle tasks with different data types, which predicts non-observed tasks via auxiliary information from associating sources.

Furthermore, in some situations, a domain can have multiple data types. For example, video data contains image and sound. How to transfer knowledge across multiple data types is worth exploring. Existing machine learning methods employ different modules to handle different data types, which leads a combination of multiple nets to tackle multiple tasks at once. This can be time and space consuming, and it requires high capacity of intelligence equipment. Therefore, new theory and models are expected to solve transfer learning with different types efficiently.

Third, existing methods assume source and target data follow stable distributions, but lack examination on stream data where source and target data may change over time in an unforeseen way. Data drift in real-world is normal, transfer learning dealing with steam data needs to be developed to keep pace with the requirements of applications in reality.

Fourth, transfer learning with scarce data needs further exploration, especially in transfer learning with heterogeneous label spaces. Existing open-set and universal transfer learning methods mainly rely on cross-entropy assumption to measure similarity between the source and target categories. However, if the source data is partial labeled, this approach can be failed due to inadequate observed information from source domain(s). In this situation, complementary learning is a possible tool to deal with unlabeled data in the source domain(s) and unknown classes in the target domain(s).

Finally, graph based transfer learning remains unsolved. Although graph convolutional networks are generally employed in existing transfer learning tasks, these methods rarely deal with real graph-structured data such as protein data. A very recent work (Levie et al., 2021) provided theory support for the transferability of graph convolutional networks, this encourages us to make further exploration in this field to achieve transfer across graph data. This can benefit the real-world applications such as those in medical field.

Bibliography

Agarwal, P., Paudel, D. P., Zaech, J.-N. & Van Gool, L., 2021, 'Unsupervised robust domain adaptation without source data', *arXiv preprint arXiv:2103.14577*.

Ahmed, S. M., Raychaudhuri, D. S., Paul, S., Oymak, S. & Roy-Chowdhury, A. K., 2021, 'Unsupervised multi-source domain adaptation without access to source data', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 10103–10112.

Ahmed, W., Morerio, P. & Murino, V., 2022, 'Cleaning noisy labels by negative ensemble learning for source-free unsupervised domain adaptation', *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1616–1625.

Ao, S., Li, X. & Ling, C. X., 2017, 'Fast generalized distillation for semi-supervised domain adaptation', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 31San Francisco, California, USA.

Arjovsky, M., Chintala, S. & Bottou, L., 2017, 'Wasserstein generative adversarial networks', *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, Australia, pp. 214–223.

Azizzadenesheli, K., Liu, A., Yang, F. & Anandkumar, A., 2019, 'Regularized learning for domain adaptation under label shifts', *International Conference on Learning Representations*, .

Baktashmotlagh, M., Chen, T. & Salzmann, M., 2022, ‘Learning to generate the unknowns as a remedy to the open-set domain shift’, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Hawaii, USA, pp. 207–216.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F. & Vaughan, J. W., 2010, ‘A theory of learning from different domains’, *Machine Learning*, vol. 79, no. 1-2, pp. 151–175.

Ben-David, S., Blitzer, J., Crammer, K. & Pereira, F., 2007, ‘Analysis of representations for domain adaptation’, *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, pp. 137–144.

Bucci, S., Borlino, F. C., Caputo, B. & Tommasi, T., 2022, ‘Distance-based hyperspherical classification for multi-source open-set domain adaptation’, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1119–1128.

Busto, P. P., Iqbal, A. & Gall, J., 2018, ‘Open set domain adaptation for image and action recognition’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 413–429.

Cao, Z., Long, M., Wang, J. & Jordan, M. I., 2018a, ‘Partial transfer learning with selective adversarial networks’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, pp. 2724–2732.

Cao, Z., Ma, L., Long, M. & Wang, J., 2018b, ‘Partial adversarial domain adaptation’, *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 135–150.

Cao, Z., You, K., Long, M., Wang, J. & Yang, Q., 2019, ‘Learning to transfer examples for partial domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 2985–2994.

Caron, M., Bojanowski, P., Joulin, A. & Douze, M., 2018, ‘Deep clustering for unsupervised learning of visual features’, *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 132–149.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P. & Joulin, A., 2020, ‘Unsupervised learning of visual features by contrasting cluster assignments’, *Advances in neural information processing systems*, vol. 33, pp. 9912–9924.

Chang, D., Sain, A., Ma, Z., Song, Y.-Z. & Guo, J., 2020, ‘Mind the gap: Enlarging the domain gap in open set domain adaptation’, *arXiv preprint arXiv:2003.03787*.

Che, X., Zuo, H., Lu, J. & Chen, D., 2021, ‘Fuzzy multi-output transfer learning for regression’, *IEEE Transactions on Fuzzy Systems*.

Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T. & Huang, J., 2019, ‘Progressive feature alignment for unsupervised domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 627–636.

Chen, G., Li, Y. & Liu, X., 2022a, ‘Transfer learning under conditional shift based on fuzzy residual’, *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 960–971.

Chen, J., Wu, X., Duan, L. & Gao, S., 2022b, ‘Domain adversarial reinforcement learning for partial domain adaptation’, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 539–553.

Chen, M., Zhao, S., Liu, H. & Cai, D., 2020, 'Adversarial-learned loss for domain adaptation', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 34New York, USA, pp. 3521–3528.

Chen, S., Hong, Z., Harandi, M. & Yang, X., 2022c, 'Domain neural adaptation', *IEEE Transactions on Neural Networks and Learning Systems*, <10.1109/TNNLS.2022.3151683>.

Chen, W., Lin, L., Yang, S., Xie, D., Pu, S., Zhuang, Y. & Ren, W., 2021, 'Self-supervised noisy label learning for source-free unsupervised domain adaptation', *arXiv preprint arXiv:2102.11614*.

Chin, T.-W., Zhang, C. & Marculescu, D., 2020, 'Improving the adversarial robustness of transfer learning via noisy feature distillation', *arXiv preprint arXiv:2002.02998*.

Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q. & Tian, Q., 2020, 'Gradually vanishing bridge for adversarial domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 12455–12464.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y. & Belongie, S., 2019, 'Class-balanced loss based on effective number of samples', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 9268–9277.

Dai, Q., Shen, X., Wu, X.-M. & Wang, D., 2019, 'Network transfer learning via adversarial domain adaptation with graph convolution', *arXiv preprint arXiv:1909.01541*.

Dai, W., Xue, G.-R., Yang, Q. & Yu, Y., 2007a, 'Transferring naive bayes classifiers for text classification', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 7Vancouver, Canada, pp. 540–545.

Dai, W., Yang, Q., Xue, G.-R. & Yu, Y., 2007b, 'Boosting for transfer learning', *Proceedings of the International Conference on Machine Learning (ICML)*, Corvallis, Oregon, USA, pp. 193–200.

Dai, W., Yang, Q., Xue, G.-R. & Yu, Y., 2008, 'Self-taught clustering', *Proceedings of the International Conference on Machine Learning (ICML)*, Helsinki, Finland, pp. 200–207.

Das, D. & Lee, C. G., 2018, 'Graph matching and pseudo-label guided deep unsupervised domain adaptation', *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, Springer, Rgides, Greece, pp. 342–352.

Davis, J. & Domingos, P., 2009, 'Deep transfer via second-order markov logic', *Proceedings of the International Conference on Machine Learning (ICML)*, Montreal, Canada, pp. 217–224.

Deng, B., Zhang, Y., Tang, H., Ding, C. & Jia, K., 2021, 'On universal black-box domain adaptation', *arXiv preprint arXiv:2104.04665*.

Deng, C., Liu, X., Li, C. & Tao, D., 2018a, 'Active multi-kernel domain adaptation for hyperspectral image classification', *Pattern Recognition*, vol. 77, pp. 306–315.

Deng, C., Xue, Y., Liu, X., Li, C. & Tao, D., 2018b, 'Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1741–1754.

Deng, Z., Choi, K.-S., Jiang, Y. & Wang, S., 2014, 'Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods', *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2585–2599.

Deng, Z., Luo, Y. & Zhu, J., 2019, 'Cluster alignment with a teacher for unsupervised domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 9944–9953.

Ding, Z., Li, S., Shao, M. & Fu, Y., 2018a, 'Graph adaptive knowledge transfer for unsupervised domain adaptation', *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 37–52.

Ding, Z., Shao, M. & Fu, Y., 2016, 'Incomplete multisource transfer learning', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 310–323.

Ding, Z., Shao, M. & Fu, Y., 2018b, 'Robust multi-view representation: A unified perspective from multi-view learning to domain adaption', *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden.

Donahue, J., Hoffman, J., Rodner, E., Saenko, K. & Darrell, T., 2013, 'Semi-supervised domain adaptation with instance constraints', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, USA, pp. 668–675.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. & Hon, H.-W., 2019, 'Unified language model pre-training for natural language understanding and generation', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, pp. 13042–13054.

Du, Y., Tan, Z., Chen, Q., Zhang, X., Yao, Y. & Wang, C., 2020, 'Dual adversarial domain adaptation', *arXiv preprint arXiv:2001.00153*.

Fang, Z., Lu, J., Liu, F., Xuan, J. & Zhang, G., 2021, 'Open set domain adaptation: Theoretical bound and algorithm', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4309–4322.

Fatras, K., Séjourné, T., Flamary, R. & Courty, N., 2021, 'Unbalanced minibatch optimal transport; applications to domain adaptation', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Virtual Online, pp. 3186–3197.

Feng, H.-Z., You, Z., Chen, M., Zhang, T., Zhu, M., Wu, F., Wu, C. & Chen, W., 2020, 'Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation', *arXiv preprint arXiv:2011.09757*.

Fu, B., Cao, Z., Long, M. & Wang, J., 2020, 'Learning to detect open classes for universal domain adaptation', *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, Virtual Online, pp. 567–583.

Fu, Y., Zhang, M., Xu, X., Cao, Z., Ma, C., Ji, Y., Zuo, K. & Lu, H., 2021, 'Partial feature selection and alignment for multi-source domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 16654–16663.

Ganin, Y. & Lempitsky, V., 2015, 'Unsupervised domain adaptation by backpropagation', *Proceedings of the International Conference on Machine Learning (ICML)*, JMLR, pp. 1180–1189.

Gong, M., Zhang, K., Huang, B., Glymour, C., Tao, D. & Batmanghelich, K., 2018, 'Causal generative domain adaptation networks', *arXiv preprint arXiv:1804.04333*.

Grant, E., Finn, C., Levine, S., Darrell, T. & Griffiths, T., 2018, 'Recasting gradient-based meta-learning as hierarchical bayes', *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, Canada.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. & Smola, A., 2006, 'A kernel method for the two-sample-problem', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, , vol. 19Vancouver, Canada, pp. 513–520.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A., 2012a, 'A kernel two-sample test', *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773.

Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K. & Sriperumbudur, B. K., 2012b, 'Optimal kernel choice for large-scale two-sample tests', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Lake Tahoe, Nevada, USA, pp. 1205–1213.

Gupta, P., Malhotra, P., Narwariya, J., Vig, L. & Shroff, G., 2020, 'Transfer learning for clinical time series analysis using deep neural networks', *Journal of Healthcare Informatics Research*, vol. 4, no. 2, pp. 112–137.

Han, H., Liu, H., Liu, Z. & Qiao, J., 2021, 'Interactive transfer learning-assisted fuzzy neural network', *IEEE Transactions on Fuzzy Systems*.

He, K., Zhang, X., Ren, S. & Sun, J., 2016, 'Deep residual learning for image recognition', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 770–778.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. & Darrell, T., 2018, 'Cycada: Cycle-consistent adversarial domain adaptation', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Vienna, Austria, pp. 1989–1998.

Hou, Y. & Zheng, L., 2020, 'Source free domain adaptation with image translation', *arXiv preprint arXiv:2008.07514*.

Hu, G., Zhang, Y. & Yang, Q., 2018a, 'Conet: Collaborative cross networks for cross-domain recommendation', *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, Turin, Italy, pp. 667–676.

Hu, J., Shen, L. & Sun, G., 2018b, 'Squeeze-and-excitation networks', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, pp. 7132–7141.

Hu, W., Miyato, T., Tokui, S., Matsumoto, E. & Sugiyama, M., 2017, 'Learning discrete representations via information maximizing self-augmented training', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Sydney, Australia, pp. 1558–1567.

Huang, J., Guan, D., Xiao, A., Lu, S. & Shao, L., 2022, 'Category contrast for unsupervised domain adaptation in visual tasks', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, Louisiana, pp. 1203–1214.

Jang, Y., Lee, H., Hwang, S. J. & Shin, J., 2019, 'Learning what and where to transfer', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, California, USA, pp. 3030–3039.

Jing, L. & Tian, Y., 2020, ‘Self-supervised visual feature learning with deep neural networks: A survey’, *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Jing, T., Liu, H. & Ding, Z., 2021, ‘Towards novel target discovery through open-set domain adaptation’, *Proceedings of the International Conference on Computer Vision (ICCV)*, Virtual online, pp. 9322–9331.

Kang, G., Jiang, L., Yang, Y. & Hauptmann, A. G., 2019, ‘Contrastive adaptation network for unsupervised domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 4893–4902.

Kang, G., Zheng, L., Yan, Y. & Yang, Y., 2018, ‘Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization’, *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 401–416.

Keneshloo, Y., Ramakrishnan, N. & Reddy, C. K., 2019, ‘Deep transfer reinforcement learning for text summarization’, *Proceedings of the SIAM International Conference on Data Mining (SDM)*, SIAM, Alberta, Canada, pp. 675–683.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C. & Krishnan, D., 2020, ‘Supervised contrastive learning’, *Advances in International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 18661–18673.

Kim, Y., Hong, S., Cho, D., Park, H. & Panda, P., 2020, ‘Progressive domain adaptation from a source pre-trained model’, *arXiv preprint arXiv:2007.01524.*

Kipf, T. N. & Welling, M., 2016, ‘Semi-supervised classification with graph convolutional networks’, *arXiv preprint arXiv:1609.02907*.

Kouw, W. M., 2018, ‘An introduction to domain adaptation and transfer learning’, *arXiv preprint arXiv:1812.11806*.

Kouw, W. M. & Loog, M., 2019, ‘A review of domain adaptation without target labels’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 766–785.

Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2012, ‘Imagenet classification with deep convolutional neural networks’, *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Lake Tahoe, Nevada, USA, pp. 1097–1105.

Kundu, J. N., Venkat, N., Babu, R. V. et al., 2020a, ‘Universal source-free domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 4544–4553.

Kundu, J. N., Venkat, N., Revanur, A., Babu, R. V. et al., 2020b, ‘Towards inheritable models for open-set domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 12376–12385.

Kurmi, V. K., Kumar, S. & Namboodiri, V. P., 2019, ‘Attending to discriminative certainty for domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 491–500.

Kurmi, V. K., Subramanian, V. K. & Namboodiri, V. P., 2021a, ‘Domain impression: A source data free domain adaptation method’, *Proceedings of the*

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Virtual online, pp. 615–625.

Kurmi, V. K., Subramanian, V. K. & Namboodiri, V. P., 2021b, 'Domain impression: A source data free domain adaptation method', *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Virtual online, pp. 615–625.

LeCun, Y., Bengio, Y. & Hinton, G., 2015, 'Deep learning', *Nature*, vol. 521, no. 7553, pp. 436–444.

Lee, J., Sattigeri, P. & Wornell, G., 2019, 'Learning new tricks from old dogs: Multi-source transfer learning from pre-trained networks', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, pp. 4372–4382.

Levie, R., Bronstein, M. M. & Kutyniok, G., 2021, 'Transferability of spectral graph convolutional neural networks', *Journal of Machine Learning Research*, vol. 22, pp. 1–59.

Li, D. & Hospedales, T., 2020, 'Online meta-learning for multi-source and semi-supervised domain adaptation', *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, Virtual online, pp. 382–403.

Li, G., Kang, G., Zhu, Y., Wei, Y. & Yang, Y., 2021a, 'Domain consensus clustering for universal domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 9757–9766.

Li, H., Jialin Pan, S., Wang, S. & Kot, A. C., 2018a, 'Domain generalization with adversarial feature learning', *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), Salt Lake City, Utah, USA, pp. 5400–5409.

Li, H., Pan, S. J., Wan, R. & Kot, A. C., 2019a, 'Heterogeneous transfer learning via deep matrix completion with adversarial kernel embedding', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 33Honolulu, Hawaii, USA, pp. 8602–8609.

Li, H., Shi, Y., Liu, Y., Hauptmann, A. G. & Xiong, Z., 2012, 'Cross-domain video concept detection: A joint discriminative and generative active learning approach', *Expert Systems with Applications*, vol. 39, no. 15, pp. 12220–12228.

Li, J., Jing, M., Su, H., Lu, K., Zhu, L. & Shen, H. T., 2021b, 'Faster domain adaptation networks', *IEEE Transactions on Knowledge and Data Engineering*, <10.1109/TKDE.2021.3060473>.

Li, J., Lu, K., Huang, Z., Zhu, L. & Shen, H. T., 2018b, 'Transfer independently together: a generalized framework for domain adaptation', *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2144–2155.

Li, J., Xu, Z., Yongkang, W., Zhao, Q. & Kankanhalli, M., 2020a, 'Gradmix: Multi-source transfer across domains and tasks', *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Colorado, USA, pp. 3019–3027.

Li, K., Lu, J., Zuo, H. & Zhang, G., 2020b, 'Multi-source domain adaptation with distribution fusion and relationship extraction', *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE, Virtual online, pp. 1–6.

Li, K., Lu, J., Zuo, H. & Zhang, G., 2021c, 'Multi-source contribution learning for domain adaptation', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, <10.1109/TNNLS.2021.3069982>.

Li, K., Lu, J., Zuo, H. & Zhang, G., 2022a, 'Dynamic classifier alignment for unsupervised multi-source domain adaptation', *IEEE Transactions on Knowledge and Data Engineering*, <10.1109/TKDE.2022.3144423>.

Li, K., Zhang, Y., Li, K., Li, Y. & Fu, Y., 2019b, 'Attention bridging network for knowledge transfer', *Proceedings of the International Conference on Computer Vision (ICCV)*, California, USA, pp. 5198–5207.

Li, L., Wan, Z. & He, H., 2021d, 'Dual alignment for partial domain adaptation', *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3404–3416.

Li, M., Zhai, Y.-M., Luo, Y.-W., Ge, P.-F. & Ren, C.-X., 2020c, 'Enhanced transport distance for unsupervised domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 13936–13944.

Li, R., Jiao, Q., Cao, W., Wong, H.-S. & Wu, S., 2020d, 'Model adaptation: Unsupervised domain adaptation without source data', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 9641–9650.

Li, W., Cao, M. & Chen, S., 2022b, 'Jacobian norm for unsupervised source-free domain adaptation', *arXiv preprint arXiv:2204.03467*.

Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L. & Huan, J., 2018c, 'Delta: Deep learning transfer using feature map with attention for convolutional networks', *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, Canada.

Li, Y., Carlson, D. E. et al., 2018d, 'Extracting relationships by multi-domain matching', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, pp. 6798–6809.

Li, Y., Yuan, L., Chen, Y., Wang, P. & Vasconcelos, N., 2021e, 'Dynamic transfer for multi-source domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 10998–11007.

Liang, J., He, R., Sun, Z. & Tan, T., 2019, 'Exploring uncertainty in pseudo-label guided unsupervised domain adaptation', *Pattern Recognition*, vol. 96, p. 106996.

Liang, J., Hu, D. & Feng, J., 2020, 'Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Virtual online, pp. 6028–6039.

Liang, J., Hu, D. & Feng, J., 2021a, 'Domain adaptation with auxiliary target domain-oriented classifier', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16632–16642.

Liang, J., Hu, D., Feng, J. & He, R., 2021b, 'Umad: Universal model adaptation under domain and category shift', *arXiv preprint arXiv:2112.08553*.

Liang, J., Hu, D., He, R. & Feng, J., 2022, 'Dine: Domain adaptation from single and multiple black-box predictors', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, .

Lifshitz, O. & Wolf, L., 2021, 'Sample selection for universal domain adaptation', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 35Virtual Online, pp. 8592–8600.

Lin, C., Zhao, S., Meng, L. & Chua, T.-S., 2020, 'Multi-source domain adaptation for visual sentiment classification', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 34New York, USA, pp. 2661–2668.

Liu, F., Lu, J. & Zhang, G., 2018, ‘Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations’, *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3555–3568.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A. & Sutherland, D. J., 2020a, ‘Learning deep kernels for non-parametric two-sample tests’, *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Virtual online, pp. 6316–6326.

Liu, F., Zhang, G. & Lu, J., 2020b, ‘Heterogeneous domain adaptation: An unsupervised approach’, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5588–5602.

Liu, F., Zhang, G. & Lu, J., 2021, ‘Multi-source heterogeneous unsupervised domain adaptation via fuzzy-relation neural networks’, *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 11, pp. 3308–3322.

Liu, H., Cao, Z., Long, M., Wang, J. & Yang, Q., 2019a, ‘Separate to adapt: Open set domain adaptation via progressive separation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 2927–2936.

Liu, H., Long, M., Wang, J. & Jordan, M. I., 2019b, ‘Towards understanding the transferability of deep representations’, *arXiv preprint arXiv:1909.12031*, <https://arxiv.org/pdf/1909.12031>.

Long, M., Cao, Y., Wang, J. & Jordan, M., 2015, ‘Learning transferable features with deep adaptation networks’, *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Lille, France, pp. 97–105.

Long, M., Wang, J., Ding, G., Sun, J. & Yu, P. S., 2013, ‘Transfer feature learning with joint distribution adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Sydney, Australia, pp. 2200–2207.

Long, M., Zhu, H., Wang, J. & Jordan, M. I., 2016, 'Unsupervised domain adaptation with residual transfer networks', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS).*

Long, M., Zhu, H., Wang, J. & Jordan, M. I., 2017, 'Deep transfer learning with joint adaptation networks', *Proceedings of the International Conference on Machine Learning (ICML)*, , vol. 70JMLR, Sydney, Australia, pp. 2208–2217.

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S. & Zhang, G., 2015, 'Transfer learning using computational intelligence: a survey', *Knowledge-Based Systems*, vol. 80, pp. 14–23.

Lu, J., Zuo, H. & Zhang, G., 2020, 'Fuzzy multiple-source transfer learning', *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3418–3431.

Luo, Y., Wang, Z., Chen, Z., Huang, Z. & Baktashmotlagh, M., 2022, 'Source-free progressive graph learning for open-set domain adaptation', *arXiv preprint arXiv:2202.06174*.

Luo, Y., Wang, Z., Huang, Z. & Baktashmotlagh, M., 2020, 'Progressive graph learning for open-set domain adaptation', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Virtual online, pp. 6468–6478.

Luo, Y., Wen, Y., Liu, T. & Tao, D., 2017, 'General heterogeneous transfer distance metric learning via knowledge fragments transfer', *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne, Australia.

Ma, A., You, F., Jing, M., Li, J. & Lu, K., 2020, 'Multi-source domain adaptation with graph embedding and adaptive label prediction', *Information Processing & Management*, vol. 57, no. 6, p. 102367.

Ma, G., Liu, F., Zhang, G. & Lu, J., 2021a, 'Learning from imprecise observations: An estimation error bound based on fuzzy random variables', *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp. 1–8.

Ma, X., Gao, J. & Xu, C., 2021b, 'Active universal domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 8968–8977.

Ma, X., Zhang, T. & Xu, C., 2019, 'Gcan: Graph convolutional adversarial network for unsupervised domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 8266–8276.

Ma, Y., Luo, G., Zeng, X. & Chen, A., 2012, 'Transfer learning for cross-company software defect prediction', *Information and Software Technology*, vol. 54, no. 3, pp. 248–256.

Maaten, L. v. d. & Hinton, G., 2008, 'Visualizing data using t-sne', *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605.

Malte, A. & Ratadiya, P., 2019, 'Evolution of transfer learning in natural language processing', *arXiv preprint arXiv:1910.07370.*

Mancini, M., Bulo, S. R., Caputo, B. & Ricci, E., 2019, 'Adagraph: Unifying predictive and continuous domain adaptation through graphs', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 6568–6577.

Mansour, Y., Mohri, M. & Rostamizadeh, A., 2009, 'Domain adaptation with multiple sources', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, pp. 1041–1048.

Matsuura, T. & Harada, T., 2020, 'Domain generalization using a mixture of multiple latent domains.', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, New York, USA, pp. 11749–11756.

Mihalkova, L. & Mooney, R. J., 2008, 'Transfer learning by mapping with minimal target data', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Workshop on Transfer Learning for Complex Tasks, Chicago, USA.

Montesuma, E. F. & Mboula, F. M. N., 2021, 'Wasserstein barycenter for multi-source domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 16785–16793.

Moon, S. & Carbonell, J. G., 2017, 'Completely heterogeneous transfer learning with attention-what and what not to transfer.', *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, , vol. 1Melbourne, Australia, pp. 1–7.

Morerio, P., Volpi, R., Ragonesi, R. & Murino, V., 2020, 'Generative pseudo-label refinement for unsupervised domain adaptation', *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Colorado, USA, pp. 3130–3139.

Müller, R., Kornblith, S. & Hinton, G., 2019, 'When does label smoothing help?', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada.

Nguyen, V.-A., Nguyen, T., Le, T., Tran, Q. H. & Phung, D., 2021, 'Stem: An approach to multi-source domain adaptation with guarantees', *Proceedings of the*

International Conference on Computer Vision (ICCV), Virtual online, pp. 9352–9363.

Niu, L., Li, W. & Xu, D., 2015, 'Multi-view domain generalization for visual recognition', *Proceedings of the International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 4193–4201.

Pan, S. J., Tsang, I. W., Kwok, J. T. & Yang, Q., 2010, 'Domain adaptation via transfer component analysis', *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210.

Pan, S. J. & Yang, Q., 2009, 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359.

Pan, Y., Yao, T., Li, Y., Ngo, C.-W. & Mei, T., 2020, 'Exploring category-agnostic clusters for open-set domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 13867–13875.

Pan, Y., Yao, T., Li, Y., Wang, Y., Ngo, C.-W. & Mei, T., 2019, 'Transferrable prototypical networks for unsupervised domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 2239–2247.

Panareda Busto, P. & Gall, J., 2017, 'Open set domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 754–763.

Park, G. Y. & Lee, S. W., 2021, 'Information-theoretic regularization for multi-source domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Virtual online, pp. 9214–9223.

Pei, Z., Cao, Z., Long, M. & Wang, J., 2018, 'Multi-adversarial domain adaptation', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 32New Orleans, Louisiana, USA, pp. 3934–3941.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K. & Wang, B., 2019a, 'Moment matching for multi-source domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 1406–1415.

Peng, X., Huang, Z., Zhu, Y. & Saenko, K., 2019b, 'Federated adversarial domain adaptation', *Proceedings of the International Conference on Learning Representations (ICLR)*, .

Qin, Y., Bruzzone, L. & Li, B., 2019, 'Tensor alignment based domain adaptation for hyperspectral image classification', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9290–9307.

Qiu, Z., Zhang, Y., Lin, H., Niu, S., Liu, Y., Du, Q. & Tan, M., 2021, 'Source-free domain adaptation via avatar prototype generation and adaptation', *arXiv preprint arXiv:2106.15326*.

Rahman, M. M., Fookes, C., Baktashmotlagh, M. & Sridharan, S., 2020, 'On minimum discrepancy estimation for deep domain adaptation', *Domain Adaptation for Visual Understanding*, Springer, pp. 81–94.

Raina, R., Battle, A., Lee, H., Packer, B. & Ng, A. Y., 2007, 'Self-taught learning: transfer learning from unlabeled data', *Proceedings of the International Conference on Machine Learning (ICML)*, Corvallis, Oregon, USA, pp. 759–766.

Redko, I., Courty, N., Flamary, R. & Tuia, D., 2019a, 'Optimal transport for multi-source domain adaptation under target shift', *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR, Naha, Okinawa, Japan, pp. 849–858.

Redko, I., Habrard, A. & Sebban, M., 2019b, 'On the analysis of adaptability in multi-source domain adaptation', *Machine Learning*, vol. 108, no. 8-9, pp. 1635–1652.

Rozantsev, A., Salzmann, M. & Fua, P., 2018, 'Residual parameter transfer for deep domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, pp. 4339–4348.

Saito, K., Kim, D., Sclaroff, S., Darrell, T. & Saenko, K., 2019, 'Semi-supervised domain adaptation via minimax entropy', *Proceedings of the International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 8050–8058.

Saito, K., Kim, D., Sclaroff, S. & Saenko, K., 2020, 'Universal domain adaptation through self supervision', *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Virtual online, pp. 16282–16292.

Saito, K. & Saenko, K., 2021, 'Ovanet: One-vs-all network for universal domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Virtual online, pp. 9000–9009.

Saito, K., Ushiku, Y. & Harada, T., 2017, 'Asymmetric tri-training for unsupervised domain adaptation', *Proceedings of the International Conference on Machine Learning (ICML)*, , vol. 70JMLR, Sydney, Australia, pp. 2988–2997.

Saito, K., Yamamoto, S., Ushiku, Y. & Harada, T., 2018, 'Open set domain adaptation by backpropagation', *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 153–168.

Settles, B., 2009, 'Active learning literature survey', Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, Madison, Wisconsin, USA.

Sharma, A., Kalluri, T. & Chandraker, M., 2021, ‘Instance level affinity-based transfer for unsupervised domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 5361–5371.

Shell, J. & Coupland, S., 2015, ‘Fuzzy transfer learning: methodology and application’, *Information Sciences*, vol. 293, pp. 59–79.

Shu, X., Qi, G.-J., Tang, J. & Wang, J., 2015, ‘Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation’, *Proceedings of the ACM International Conference on Multimedia (ICME)*, Torino, Italy, pp. 35–44.

Shu, Y., Cao, Z., Wang, C., Wang, J. & Long, M., 2021, ‘Open domain generalization with domain-augmented meta-learning’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9624–9633.

Sohn, K., Shang, W., Yu, X. & Chandraker, M., 2019, ‘Unsupervised domain adaptation for distance metric learning’, *International Conference on Learning Representations*, .

Sun, B. & Saenko, K., 2016, ‘Deep coral: Correlation alignment for deep domain adaptation’, *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, Amsterdam, The Netherlands, pp. 443–450.

Sun, R., Zhu, X., Wu, C., Huang, C., Shi, J. & Ma, L., 2019, ‘Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 4360–4369.

Tan, B., Song, Y., Zhong, E. & Yang, Q., 2015, ‘Transitive transfer learning’, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Sydney, Australia, pp. 1155–1164.

Tan, B., Zhang, Y., Pan, S. J. & Yang, Q., 2017, 'Distant domain transfer learning', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 300San Francisco, California, USA, pp. 301–302.

Tang, H., Chen, K. & Jia, K., 2019, 'Unsupervised domain adaptation via structurally regularized deep clustering', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, California, USA, pp. 8725–8735.

Tang, J., Shu, X., Li, Z., Qi, G.-J. & Wang, J., 2016, 'Generalized deep transfer networks for knowledge propagation in heterogeneous domains', *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 4, pp. 1–22.

Tian, L., Tang, Y., Hu, L., Ren, Z. & Zhang, W., 2020a, 'Domain adaptation by class centroid matching and local manifold self-learning', *IEEE Transactions on Image Processing*, vol. 29, pp. 9703–9718.

Tian, Q., Ma, C., Cao, M. & Chen, S., 2020b, 'Domain adaptation through transferring both the source-knowledge and target-relatedness simultaneously', *arXiv preprint arXiv:2003.08051*.

Tian, Q., Sun, H., Ma, C., Cao, M., Chu, Y. & Chen, S., 2021, 'Heterogeneous domain adaptation with structure and classification space alignment', *IEEE Transactions on Cybernetics*.

Tzeng, E., Hoffman, J., Saenko, K. & Darrell, T., 2017, 'Adversarial discriminative domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, pp. 7167–7176.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K. & Darrell, T., 2014, ‘Deep domain confusion: Maximizing for domain invariance’, *arXiv preprint arXiv:1412.3474*.

Vapnik, V. & Vapnik, V., 1998, ‘Statistical learning theory’, *New York*, vol. 1, p. 624.

Vapnik, V. N., 1999, ‘An overview of statistical learning theory’, *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999.

Venkateswara, H., Eusebio, J., Chakraborty, S. & Panchanathan, S., 2017, ‘Deep hashing network for unsupervised domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA.

Wang, H., Xu, M., Ni, B. & Zhang, W., 2020a, ‘Learning to combine: Knowledge aggregation for multi-source domain adaptation’, *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, pp. 727–744.

Wang, J., Chen, Y., Feng, W., Yu, H., Huang, M. & Yang, Q., 2020b, ‘Transfer learning with dynamic distribution adaptation’, *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 1, pp. 1–25.

Wang, J. et al., n.d., ‘Everything about transfer learning and domain adapation’, http://transferlearning.xyz.

Wang, R., Wu, Z., Weng, Z., Chen, J., Qi, G.-J. & Jiang, Y.-G., 2022, ‘Cross-domain contrastive learning for unsupervised domain adaptation’, *IEEE Transactions on Multimedia*, <10.1109/TMM.2022.3146744>.

Wang, X., Huang, T.-K. & Schneider, J., 2014, ‘Active transfer learning under model shift’, *Proceedings of the International Conference on Machine Learning (ICML)*, Beijing, China, pp. 1305–1313.

Wang, X., Jin, Y., Long, M., Wang, J. & Jordan, M. I., 2019a, ‘Transferable normalization: Towards improving transferability of deep neural networks’, *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, , vol. 32Vancouver, Canada, pp. 1953–1963.

Wang, X., Li, L., Ye, W., Long, M. & Wang, J., 2019b, ‘Transferable attention for domain adaptation’, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 33Honolulu, Hawaii, USA, pp. 5345–5352.

Wang, Y. G., Li, M., Ma, Z., Montufar, G., Zhuang, X. & Fan, Y., 2019c, ‘Haarpooling: Graph pooling with compressive haar basis’, *arXiv preprint arXiv:1909.11580.*

Wei, Y., Zhang, Y., Huang, J. & Yang, Q., 2018, ‘Transfer learning via learning to transfer’, *Proceedings of the International Conference on Machine Learning (ICML)*, , vol. 80PMLR, Stockholm, Sweden, pp. 5085–5094.

Wei, Y., Zhu, Y., Leung, C. W.-k., Song, Y. & Yang, Q., 2016, ‘Instilling social to physical: Co-regularized heterogeneous transfer learning’, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 30Phoenix, Arizona, USA.

Weiss, K., Khoshgoftaar, T. M. & Wang, D., 2016, ‘A survey of transfer learning’, *Journal of Big Data*, vol. 3, no. 1, p. 9.

Wen, J., Greiner, R. & Schuurmans, D., 2020, ‘Domain aggregation networks for multi-source domain adaptation’, *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Virtual online, pp. 10214–10224.

Wen, J., Liu, R., Zheng, N., Zheng, Q., Gong, Z. & Yuan, J., 2019, 'Exploiting local feature patterns for unsupervised domain adaptation', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 33Honolulu, Hawaii, USA, pp. 5401–5408.

Wu, K., Shi, Y., Han, Y., Shao, Y., Li, B. & Tian, Q., 2021, 'Domain adaptation without model transferring', *arXiv preprint arXiv:2107.10174*.

Wu, Q., Wu, H., Zhou, X., Tan, M., Xu, Y., Yan, Y. & Hao, T., 2017, 'Online transfer learning with multiple homogeneous or heterogeneous sources', *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1494–1507.

Xia, H., Zhao, H. & Ding, Z., 2021, 'Adaptive adversarial network for source-free domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 9010–9019.

Xiao, M. & Guo, Y., 2014, 'Feature space independent semi-supervised domain adaptation via kernel matching', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 54–66.

Xiao, N. & Zhang, L., 2021, 'Dynamic weighted learning for unsupervised domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 15242–15251.

Xie, L., Deng, Z., Xu, P., Choi, K.-S. & Wang, S., 2018, 'Generalized hidden-mapping transductive transfer learning for recognition of epileptic electroencephalogram signals', *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2200–2214.

Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q. & Zhang, W., 2020a, 'Adversarial domain adaptation with domain mixup', *Proceedings of the AAAI*

Conference on Artificial Intelligence (AAAI), , vol. 34New York, USA, pp. 6502–6509.

Xu, P., Deng, Z., Wang, J., Zhang, Q., Choi, K.-S. & Wang, S., 2021a, 'Transfer representation learning with tsk fuzzy system', *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 649–663.

Xu, R., Chen, Z., Zuo, W., Yan, J. & Lin, L., 2018, 'Deep cocktail network: Multi-source unsupervised domain adaptation with category shift', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, pp. 3964–3973.

Xu, R., Li, G., Yang, J. & Lin, L., 2019a, 'Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 1426–1435.

Xu, R., Li, G., Yang, J. & Lin, L., 2019b, 'Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, California, USA, pp. 1426–1435.

Xu, R., Liu, P., Wang, L., Chen, C. & Wang, J., 2020b, 'Reliable weighted optimal transport for unsupervised domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 4394–4403.

Xu, Y., Chen, L., Duan, L., Tsang, I. W. & Luo, J., 2021b, 'Open set domain adaptation with soft unknown-class rejection', *IEEE Transactions on Neural Networks and Learning Systems*, <10.1109/TNNLS.2021.3105614>.

Xu, Y., Kan, M., Shan, S. & Chen, X., 2022, ‘Mutual learning of joint and separate domain alignments for multi-source domain adaptation’, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1890–1899.

Xu, Y. & Yan, H., 2022, ‘Cycle-reconstructive subspace learning with class discriminability for unsupervised domain adaptation’, *Pattern Recognition*, p. 108700.

Yan, Y., Wu, Q., Tan, M., Ng, M. K., Min, H. & Tsang, I. W., 2017, ‘Online heterogeneous transfer by hedge ensemble of offline and online decisions’, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3252–3263.

Yang, B. & Yuen, P. C., 2019, ‘Cross-domain visual representations via unsupervised graph alignment’, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 33Honolulu, Hawaii, USA, pp. 5613–5620.

Yang, L., Hanneke, S. & Carbonell, J., 2013, ‘A theory of transfer learning with applications to active learning’, *Machine Learning*, vol. 90, no. 2, pp. 161–189.

Yang, S., Kim, Y., Jung, D. & Kim, C., 2020a, ‘Partial domain adaptation using graph convolutional networks’, *arXiv preprint arXiv:2005.07858*.

Yang, S., van de Weijer, J., Herranz, L., Jui, S. et al., 2021a, ‘Exploiting the intrinsic neighborhood structure for source-free domain adaptation’, *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Virtual online.

Yang, S., Wang, Y., van de Weijer, J., Herranz, L. & Jui, S., 2021b, ‘Casting a bait for offline and online source-free domain adaptation’, *arXiv preprint arXiv:2010.12427*, vol. 1, no. 2, p. 3.

Yang, S., Wang, Y., van de Weijer, J., Herranz, L. & Jui, S., 2021c, 'Generalized source-free domain adaptation', *Proceedings of the International Conference on Computer Vision (ICCV)*, Virtual online, pp. 8978–8987.

Yang, S., Wang, Y., Wang, K., Jui, S. & van de Weijer, J., 2022, 'One ring to bring them all: Towards open-set recognition under domain shift', *arXiv preprint arXiv:2206.03600.*

Yang, X., Deng, C., Liu, T. & Tao, D., 2020b, 'Heterogeneous graph attention network for unsupervised multiple-target domain adaptation', *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Yao, T., Pan, Y., Ngo, C.-W., Li, H. & Mei, T., 2015, 'Semi-supervised domain adaptation with subspace learning for visual recognition', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 2142–2150.

Yao, Y., Zhang, Y., Li, X. & Ye, Y., 2020, 'Discriminative distribution alignment: A unified framework for heterogeneous domain adaptation', *Pattern Recognition*, vol. 101, p. 107165.

Ye, Y., Huang, Z., Pan, T., Li, J. & Shen, H. T., 2021, 'Reducing bias to source samples for unsupervised domain adaptation', *Neural Networks*, vol. 141, pp. 61–71.

Yin, Y., Yang, Z., Hu, H. & Wu, X., 2022, 'Universal multi-source domain adaptation for image classification', *Pattern Recognition*, vol. 121, p. 108238.

You, K., Long, M., Cao, Z., Wang, J. & Jordan, M. I., 2019, 'Universal domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 2720–2729.

Yu, C., Wang, J., Chen, Y. & Huang, M., 2019a, ‘Transfer learning with dynamic adversarial adaptation network’, *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Beijing, China, pp. 778–786.

Yu, C., Wang, J., Chen, Y. & Wu, Z., 2019b, ‘Accelerating deep unsupervised domain adaptation with transfer channel pruning’, *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE, Budapest, Hungary, pp. 1–8.

Yu, F., Zhao, J., Gong, Y., Wang, Z., Li, Y., Yang, F., Dong, B., Li, Q. & Zhang, L., 2019c, ‘Annotation-free cardiac vessel segmentation via knowledge transfer from retinal images’, *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, Shenzhen, China, pp. 714–722.

Yu, H., Hu, M. & Chen, S., 2018, ‘Multi-target unsupervised domain adaptation without exactly shared categories’, *arXiv preprint arXiv:1809.00852*.

Yuan, J., Hou, F., Du, Y., Shi, Z., Geng, X., Fan, J. & Rui, Y., 2022, ‘Self-supervised graph neural network for multi-source domain adaptation’, *arXiv preprint arXiv:2204.05104*.

Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K. & Vincentelli, A. S., 2021, ‘Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation’, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual online, pp. 13834–13844.

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T. & Saminger-Platz, S., 2017, ‘Central moment discrepancy (cmd) for domain-invariant representation learning’, *arXiv preprint arXiv:1702.08811*.

Zhang, C. & Zhao, Q., 2021, 'Attention guided for partial domain adaptation', *Information Sciences*, vol. 547, pp. 860–869.

Zhang, J., Ding, Z., Li, W. & Ogunbona, P., 2018, 'Importance weighted adversarial nets for partial domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake, USA, pp. 8156–8164.

Zhang, J., Li, W. & Ogunbona, P., 2017, 'Joint geometrical and statistical alignment for visual domain adaptation', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu. Hawaii, USA, pp. 1859–1867.

Zhang, J., Zhou, W., Chen, X., Yao, W. & Cao, L., 2019a, 'Multi-source selective transfer framework in multi-objective optimization problems', *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 424–438.

Zhang, L., 2019, 'Transfer adaptation learning: A decade survey', *arXiv preprint arXiv:1903.04687*.

Zhang, Y., Liu, F., Fang, Z., Yuan, B., Zhang, G. & Lu, J., 2020, 'Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation', *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Yokohama, Japan, pp. 2526–2532.

Zhang, Y., Liu, T., Long, M. & Jordan, M., 2019b, 'Bridging theory and algorithm for domain adaptation', *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, Long beach, USA, pp. 7404–7413.

Zhang, Y. & Yang, Q., 2017, 'A survey on multi-task learning', *arXiv preprint arXiv:1707.08114*.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P. & Gordon, G. J., 2018, ‘Adversarial multiple source domain adaptation’, *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, pp. 8559–8570.

Zhao, J., Chen, Y. & Zhang, W., 2019a, ‘Differential privacy preservation in deep learning: Challenges, opportunities and solutions’, *IEEE Access*, vol. 7, pp. 48901–48911.

Zhao, P., Hoi, S. C., Wang, J. & Li, B., 2014, ‘Online transfer learning’, *Artificial Intelligence*, vol. 216, pp. 76–102.

Zhao, S., Li, B., Xu, P., Yue, X., Ding, G. & Keutzer, K., 2021, ‘Madan: multi-source adversarial domain aggregation network for domain adaptation’, *International Journal of Computer Vision*, vol. 9, no. 129, pp. 2399–2424.

Zhao, S., Li, B., Yue, X., Gu, Y., Xu, P., Hu, R., Chai, H. & Keutzer, K., 2019b, ‘Multi-source domain adaptation for semantic segmentation’, *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, pp. 7285–7298.

Zhao, S., Wang, G., Zhang, S., Gu, Y., Li, Y., Song, Z., Xu, P., Hu, R., Chai, H. & Keutzer, K., 2020a, ‘Multi-source distilling domain adaptation’, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 34New York, USA, pp. 12975–12983.

Zhao, S., Yue, X., Zhang, S., Li, B., Zhao, H., Wu, B., Krishna, R., Gonzalez, J. E., Sangiovanni-Vincentelli, A. L., Seshia, S. A. et al., 2020b, ‘A review of single-source deep unsupervised visual domain adaptation’, *IEEE Transactions on Neural Networks and Learning Systems*.

Zheng, V. W., Pan, S. J., Yang, Q. & Pan, J. J., 2008, 'Transferring multi-device localization models using latent multi-task learning.', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 8Chicago, USA, pp. 1427–1432.

Zhou, J. T., Pan, S. J., Tsang, I. W. & Ho, S.-S., 2016, 'Transfer learning for cross-language text categorization through active correspondences construction', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 30Phoenix, Arizona, USA.

Zhou, L., Ye, M., Zhang, D., Zhu, C. & Ji, L., 2021, 'Prototype-based multisource domain adaptation', *IEEE Transactions on Neural Networks and Learning Systems*.

Zhu, Y., Zhuang, F. & Wang, D., 2019a, 'Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources', *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, , vol. 33Honolulu, Hawaii, USA, pp. 5989–5996.

Zhu, Y., Zhuang, F., Wang, J., Chen, J., Shi, Z., Wu, W. & He, Q., 2019b, 'Multi-representation adaptation network for cross-domain image classification', *Neural Networks*, vol. 119, pp. 214–221.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. & He, Q., 2020, 'A comprehensive survey on transfer learning', *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76.

Zuo, H., Lu, J., Zhang, G. & Liu, F., 2018a, 'Fuzzy transfer learning using an infinite gaussian mixture model and active learning', *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 2, pp. 291–303.

Zuo, H., Lu, J., Zhang, G. & Pedrycz, W., 2018b, 'Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces', *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 2, pp. 348–361.

Zuo, H., Zhang, G., Pedrycz, W., Behbood, V. & Lu, J., 2016, 'Fuzzy regression transfer learning in takagi–sugeno fuzzy models', *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1795–1807.

Zuo, H., Zhang, G., Pedrycz, W., Behbood, V. & Lu, J., 2017, 'Granular fuzzy regression domain adaptation in takagi–sugeno fuzzy models', *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 847–858.