Interpretable Domain-Aware Learning for Neuroimage Classification

by

Shuo Zhou

Supervisor: Dr Haiping Lu

Department of Computer Science

The University of Sheffield

This thesis is submitted for the degree of
Doctor of Philosophy

Feburary 28, 2022

To my lovely wife Sisi Yao, and a real artificial intelligence, Yijun (Ole) Zhou.

Declaration

All sentences or passages quoted in this thesis from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this thesis have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in the degree examination as a whole.

Name: Shuo Zhou

Signature: Shuo Zhou

Date: Feburary 28, 2022

Abstract

In this thesis, we propose three interpretable domain-aware machine learning approaches to analyse large-scale neuroimaging data from multiple domains, e.g. multiple centres and/or demographic groups. We focus on two questions: how to learn general patterns across domains, and how to learn domain-specific patterns.

Our first approach develops a feature-classifier adaptation framework for semi-supervised domain adaptation on brain decoding tasks. Based on this empirical study, we derive a dependence-based generalisation bound to guide the design of domain-aware learning algorithms. This theoretical result leads to the next two approaches. The covariate-independence regularisation approach is for learning domain-generic patterns. Incorporating hinge and least squares loss generates two covariate-independence regularised classifiers, whose superiority are validated by the experimental results on brain decoding tasks for unsupervised multi-source domain adaptation. The covariate-dependent learning approach is for learning domain-specific patterns, which can learn gender-specific patterns of brain lateralisation via employing the logistic loss.

Interpretability is often essential for neuroimaging tasks. Therefore, all three domain-aware learning approaches are primarily designed to produce linear, interpretable models. These domain-aware learning approaches offer feasible ways to learn interpretable general or specific patterns from multi-domain neuroimaging data for neuroscientists to gain insights. With source code released on GitHub, this work will accelerate data-driven neuroimaging studies and advance multi-source domain adaptation research.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Dr. Haiping Lu, who has been my supervisor since my master dissertation project in 2016. I was provided valuable suggestions, enormous support and continuous encouragement during my Ph.D study. His knowledge, vision, and attitude to research provide me lifetime benefits. I am very grateful to Dr. Mauricio A. Álvarez and Prof. Richard Clayton, who have always been supportive as my advisors. Also, I would like to thank our collaborating neuroscientists, Dr. Christopher Cox and Prof. Gaolang Gong, who helped me to deepen my understanding of neuroimaging analysis and neuroscience.

Special thanks to all the members of the machine learning lab at 136, Regent Court. Thanks for the wonderful memories in the past four years and the support and encouragement from each other, especially during the COVID pandemic. I would also like to thank the staffs and students of State Key Laboratory of Cognitive Neuroscience and Learning, Beijing Normal University, for their hospitality during my visit. Especially Mr. Junhao Luo, who was born to be a professor. All the best for the thesis writing and looking forward to our future collaborations.

Finally, special gratitude is given to my family, my wife Sisi, and both Sisi's and my parents, for their love, consideration, and support. And of course, little Ole, who joined our family during my Ph.D study, "You make me happy when skies are grey".

Thanks to all the people who have helped and encouraged me. I could never accomplish all these achievements without you.

Contents

List of Figures

List of Tables

Symbols and Notations

Notations

Scalers	Lower case letters, e.g., x
Vectors	Lower case bold letters, e.g, $\mathbf{x}$
Constant	Upper case letters, e.g., C
Matrices	Upper case bold letters, e.g., $\mathbf{X}$
Spaces	Upper case calligraphic letters, e.g., $\mathcal{X}$

Pre-Defined Symbols

c	Target concept
d	Input feature dimension
$\mathbb{E}$	Expectation
f or h	Classifier, hypothesis, labelling function, or machine learning model
$\mathbf{H}$	Centring matrix
$\mathcal{H}$	Hypothesis space, or reproducing kernel Hilbert space (RKHS)
$\mathbf{I}$	Identity matrix
$\mathbf{K}$	Kernel matrix
$k(\cdot, \cdot)$	Kernel function, e.g., linear, Gaussian, polynomial
m_t, m_s	Target/source sample size
$\mathrm{O}(\cdot)$	Computational complexity
$\mathbb{P}$	Probability
$\mathrm{R}(\cdot)$ / $\hat{\mathrm{R}}(\cdot)$	Risk (error)/empirical risk
$\mathfrak{R}(\cdot)$	Rademacher complexity
$\mathrm{tr}(\cdot)$	Trace function
$\mathrm{VC}(\cdot)$	VC-dimension
$\mathbf{w}$	Model coefficient vector
$\mathbf{x}$	Input vector
$\mathbf{X}$	Input data matrix
$\mathbf{X}^t$, $\mathbf{X}^s$	Target/source data matrix
$\mathcal{X}$	Input space
y	Label or output
$\mathbf{Y}$	Output matrix
$\mathcal{Y}$	Output space

Chapter 1

Introduction

The subject of this thesis is learning patterns from neuroimaging data with multiple domains automatically. That is, using computational algorithms to convert the underlying patterns of neuroimages into knowledge, or expertise. The learnt knowledge or expertise are usually in the form of statistical models. Due to the high cost of data acquisition, the sample sizes of neuroimaging datasets are usually small but with high feature dimensions. The emergence of public neuroimaging datasets opens a door to analysing brain imaging data across datasets. However, generalisation is still a challenging problem. On the other hand, for some neuroscience research problems, neuroimaging data need to be viewed as different blocks/groups, e.g. understanding the patterns across genders, age groups, or patient/control. In this thesis, we formulate these cross datasets or groups, gender, or age analysis as a "multi-domain" problem and tackle domain-aware learning approaches. More importantly, the proposed approaches are primarily designed to be linear for interpretability. By visualising the linear model coefficients in feature space, we validated existing findings of neuroscience and identified new, meaningful patterns to offer additional insights.

1.1 Machine Learning and Multi-Domain Problems

Machine learning is a kind of technique to detect underlying patterns automatically from training data (Shalev-Shwartz & Ben-David, 2014). The learning tasks can vary, e.g., classification, regression, and clustering. The focus of this thesis is classification of neuroimaging

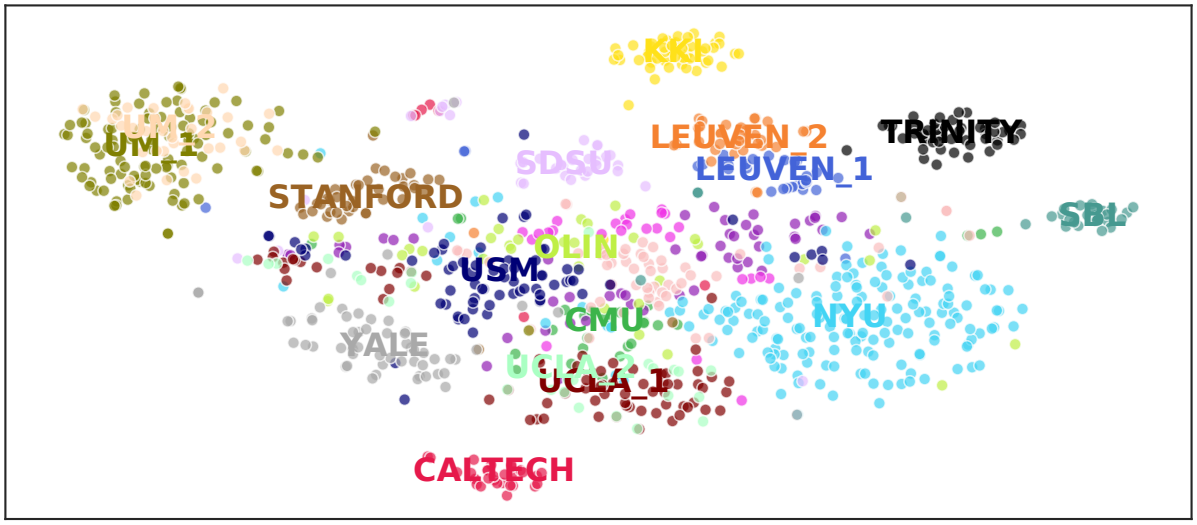


Figure 1.1: Visualisation of the resting-fMRI data from ABIDE (Craddock et al., 2013) dataset. Different colours denote different centres where the samples were acquired. In total there are 20 centres. Image source: Kunda et al. (2020).

data analysis. The objective of classification is assigning an input vector $\mathbf{x}$ to one of K discrete classes y, where $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, $|\mathcal{Y}| = K$, and $\mathcal{X}, \mathcal{Y}$ are input and output spaces, respectively. Conventionally, each input instance can only be assigned to only one class. It is also the basic problem setting of this thesis, and multi-output classification problems, i.e. an instance can have more than one label, are not considered here.

For classification problems, what needs to be learnt is a labelling function (or a hypothesis) for assigning an input to a class, i.e., $y = f(\mathbf{x})$, where f is the labelling function. Usually the labelling function f is learnt by fitting the data of the labelled training set, and its performance is evaluated by how well it can assign unseen samples (test data) to correct classes. In the most common scenario, the training and testing sample sets are assumed drawn from the same distribution, where the performance on test sets (generalisability) is guaranteed according to the statistical learning theory (Vapnik & Chervonenkis, 1971). However in practice, the distributions of training / testing input data can be changed in some way. For example, for neuroimaging data acquired from different sites, the distributions can be shifted due to the differences in devices, physical environments, etc. Figure 1.1 shows an example of multi-centre resting-fMRI dataset ABIDE (Craddock et al., 2013). From the visualisation, it can be observed that samples from different centres are usually in different

clusters. Traditional methods can still be applied to such problems by either ignoring the potential relationships or differences between different sources, or only taking one subgroup of data to perform machine learning. However, this could lead to a degeneration in learning performance, especially when the number of total available training samples is small.

Domain adaptation (DA) is an emerging machine learning technique to solve the problem of generalisability. DA is a branch of transfer learning, where data distribution is assumed to change across sample sets, and each distribution is viewed as a domain. In a domain adaptation problem, the learning tasks are the same cross domains (Redko et al., 2020), the target dataset (for making prediction) is usually called the target domain, and the other sources leveraged for training are called source domains. From the theoretical perspective, Ben-David et al. (2010) pointed out that the expected risk (error) of a labelling function f on target is bounded by the error on sources plus divergence between domains. Therefore in practice, domain adaptation methods adopted distribution divergence as regularisations to perform DA, including single-source (Pan et al., 2011; Long et al., 2013a,b; Wang et al., 2018), and multi-source (Guo et al., 2018; Zhao et al., 2018; Dou et al., 2019; Zhu et al., 2019; Peng et al., 2019) DA studies. The experimental results in these works indicate domain adaptation is a promising approach for multi-domain data analysis.

Contrary to domain adaptation that aims at improving generalisability, learning domain-specific patterns is another research direction. For this problem, separating general and specific features (Lock et al., 2013; Zhou et al., 2015) is the most popular solution, which assumes the multi-domain data is the mixture of general and specific features plus noise. The most important step of feature separation is extracting general features across domains, which is similar to domain adaptation approaches. Then specific features can be obtained by deducting general features from cleaned / denoised data.

1.2 Background of Neuroimaging Data

Neuroimaging is a set of techniques for investigating human brain structures or functions. This thesis will focus on the analysis of functional imaging, i.e., functional magnetic resonance imaging (fMRI) (Ogawa et al., 1990). Functional neuroimaging analysis highly relies on

statistical approaches, due to lacking of structural information and therefore not readable by human visually. This section will introduce the basic background of functional neuroimaging data and popular methods of analysis.

1.2.1 Functional MRI (fMRI)

Functional magnetic resonance imaging (fMRI) is a medical imaging technique that records the Blood Oxygenation Level Dependent (BOLD) signal caused by changes in blood flow Ogawa et al. (1990). Typically, an fMRI sequence is composed of MRI volumes sampled every few seconds, where each MRI volume has over 100,000 voxels and each voxel represents the aggregate activity within a small volume ($2 - 3mm^3$). fMRI can measure the neural activity associated with multiple cognitive behaviours and examine brain functions in healthy relatives to disordered individuals.

There are two distinct types of fMRI data: **task-based** fMRI (or model-based fMRI) and **resting-state** fMRI (Friston et al., 1998). The task-based fMRI which is based on designed experiments or models is aiming at measuring the brain response to a stimulus/event (or event block). Neuroscientists need to design cognitive experiments associated with specific brain functions of interest for data collection (Brodersen et al., 2011). The fMRI data are recorded when the participants are performing the cognitive tasks according to the instructions. By contrast, resting-state fMRI reflects the "finger-print" (Finn et al., 2015) of one subject's brain, which is model-free so participants do not need to do some tasks during the data collection (Fox & Raichle, 2007).

1.2.2 Machine Learning in fMRI Analysis

Distinguishing functional brain activities can be framed as classification problems and solved with machine learning techniques. The following will introduce two most common applications: classifying cognitive conditions (tasks) (Norman et al., 2006; Tong & Pratte, 2012) and clinical populations (Cheng et al., 2015c; Gheiratmand et al., 2017).

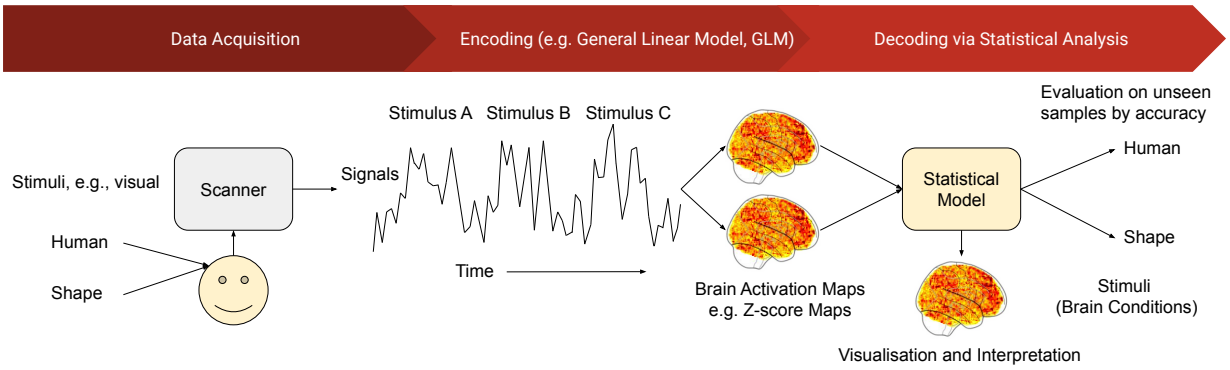


Figure 1.2: An example of the brain decoding process. In a cognitive experiment, participants are usually required to perform some tasks that react to the stimulus designed by neuroscientists. The corresponding brain signals will be recorded by the scanners and encoded as statistical brain activation maps. The objective of brain decoding is using machine learning approaches to decode (classify) the given brain activation maps into correct brain conditions (stimuli), to help neuroscientists to understand human brain functions.

Brain Decoding with Task-Based fMRI

Task-based fMRI images are usually pre-processed, and modelled through a General Linear Model (GLM). The resulting statistical brain activation maps represent the impact of different experimental stimuli (conditions) on fMRI signals. This process is usually called encoding.

Traditional analysis for brain activation maps relies on univariate statistical tests (Friston et al., 1994, 1995; Worsley & Friston, 1995). This approach can help neuroscientists to identify the activated voxels corresponding to experimental stimuli. However, the multivariate structure of fMRI data cannot be taken into account.

In the recent two decades, a brain decoding (or "brain reading") approach (Cox & Savoy, 2003) has been proposed by using machine learning methods to decode the neural encodings. This multivariate pattern recognition framework can provide more sensitive analysis than the univariate methods. Many algorithms have been applied for classifying and regressing brain activation images, such as Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Lasso, and Elastic net (Mandelkow et al., 2016; LaConte et al., 2005; Toiviainen et al., 2014; Ng et al., 2010). Figure 1.2 depicts the whole process of brain decoding.

Brain Network Modelling and Classification

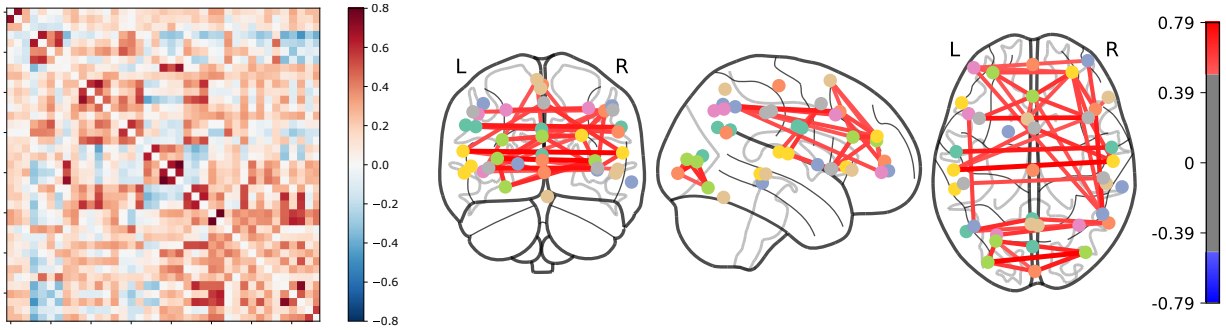
Brain network connectome (functional connectivity) analysis is another popular feature extraction method for fMRI data, especially for resting-state fMRI time-series. A functional connectome is a set of connections representing brain interactions between regions. The regions used to extract the signal can be defined by brain parcellations, which are pre-defined brain atlases with regions of interest (ROI). Alternatively, these regions can be defined by performing independent component analysis (ICA) (Hyvärinen & Oja, 2000).

A functional connectome can be represented by a number of regions $\times$ number of regions square matrix, such as a correlation matrix (shown in Fig. 1.3a), and can also be seen as a graph: a set of nodes, connected by edges. In fMRI analysis, these nodes are brain regions, and the edges capture interactions between them, this graph is a functional connectome (shown in Fig. 1.3b). There are several different kind of correlation to measure the interaction between brain regions, such as covariance, partial covariance, group-sparse covariance (Smith et al., 2011; Varoquaux et al., 2010b) and tangent connectome (Varoquaux et al., 2010a).

With the brain connectivity measurement, the brain networks can be further represented as the vectorised upper (or lower) triangle of the connectome matrix, where each feature represents the connectivity between two brain regions. Since resting-state reflects natural patterns of one's brain, brain network features can be used to investigate some brain disorders. Especially for the diseases that cannot be identified directly from brain structures, functional neuroimaging plays an important role in understanding the cause and localising biomarkers. In such research areas, machine learning classifiers have been widely used, such as classifying patients and health controls, to investigate brain disorders in a data-driven way (Kunda et al., 2020; Schirmer et al., 2021).

1.2.3 Challenges

The primary challenge with using an fMRI dataset to train a machine learning model for both task-based and resting-state fMRI analysis suffers from the "curse of dimensionality". For example in brain decoding tasks, the number of training samples available for machine learning are typically less than one hundred ($\sim$100) per brain state, which is much smaller



(a) Brain network as a matrix.

(b) Brain network visualisation on a reference atlas.

Figure 1.3: Examples of visualised brain networks, where each feature represents the connectivity between two brain regions. The figures are generated by the Python library Nilearn (Abraham et al., 2014).

compared to the feature (voxel) space (∼100,000). Similar in resting-state fMRI analysis, using hundreds of brain regions can result in over ten thousands of connectivity features, e.g, over 20,000 features can be extracted with one of the most popular atlas CC200 (200 regions), while the samples for training are usually less than one thousand. This makes accurate "out of sample" prediction a challenging problem.

Traditionally, this challenge is dealt with by pre-selecting voxels that belong to regions of interest (ROIs) based on prior work and established knowledge of domain experts Poldrack (2007), or by performing a "searchlight" analysis (Kriegeskorte et al., 2006). While making the problem computationally more tractable, it may ignore a significant portion of information in fMRI, and miss potentially valid and superior solutions in the first place. Recently, studies of *whole-brain fMRI* are becoming increasingly popular (Allen et al., 2014; Gonzalez-Castillo et al., 2015; Vu et al., 2015). This approach not only broadens the scope of potentially important differences between cognitive states, but also lifts the need for a priori assumptions about which parts of the brain are most relevant. Whole-brain fMRI analysis can take all available information into account in a more *data-driven* workflow, however the major bottleneck remains the size discrepancy between training example and features.

Furthermore, neuroimaging analysis requires not only accurate models, but also the capability of model interpreting and understanding which brain areas are important for classification. As mentioned in Section 1.2.1, functional neuroimaging analysis is important for neu-

roscientists to understand human brain functions and highly relies on statistical approaches. Linear models have advantages in interpretability however they may not be able to capture the non-linear correlations between input and output, which makes accurate prediction a more challenging problem.

While any individual fMRI dataset provides only a few training examples, growing public data repositories collectively contain much more training examples, such as OpenNeuro[1] (Gorgolewski et al., 2017) and the Human Connectome Project[2] (HCP) (Smith et al., 2013). Although every neuroimaging experiment is importantly different, many recruit similar sets of cognitive functions. If training examples from related, pre-existing datasets can be leveraged to improve the performance and interpretability of decoding models, it would unlock immense latent power in big data resources that already exist in the neuroimaging community. These data from multiple domains enable multi-domain learning studies on neuroimaging data analysis, which is the primary focus of this thesis.

1.3 Hypothesis and Research Questions

Hypothesis: For multi-domain neuroimaging data, it is hypothesised there exist general and specific patterns. Based on the hypothesis, by letting machine learning algorithms be aware of the domain information, this thesis aims to answer the two following research questions:

- **Q1:** How to learn general patterns across different domains.
- **Q2:** How to learn patterns specifically for a domain.

1.4 Organisation

To answer the aforementioned two questions, the rest of this thesis will be organised into six chapters. Q1 will be mainly targeted by Chapter 3 and 5, Q2 will be targeted by Chapter 6, and a theoretical support can be used for both Q1 and Q2 will be presented in Chapter

[1] `https://openneuro.org/`
[2] `https://www.humanconnectome.org/`

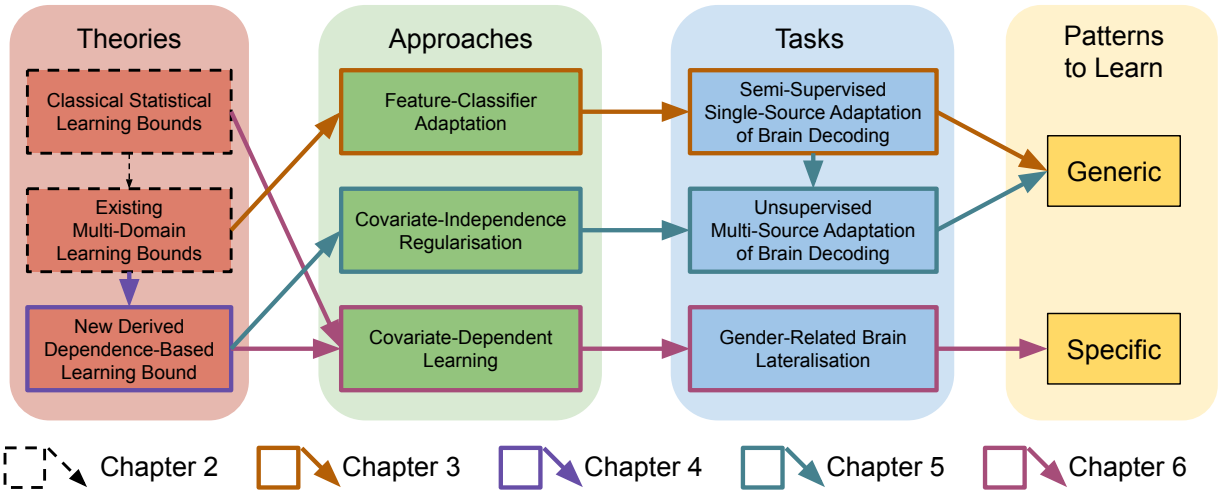


Figure 1.4: Organisation of this thesis (Chapter 2 $\sim$ 6), where the pathways of each chapter are denoted by different colours and/or styles.

4. Figure 1.4 depicts the relationships of these chapters. Detailed summaries for each of the following chapter are listed below:

Chapter 2 Fundamentals of Learning from Multiple Domains. In this chapter, the background knowledge required for later theory and algorithm development will be covered. This includes: 1) Probably Approximately Correct (PAC) learning model based statistical theory of generalisation. The generalisation bounds for standard machine learning scenarios will be introduced first and then extended to multi-domain learning problems. 2) A summary of popular multi-domain learning methods with their applications, which covers both domain adaptation and common individual feature separation algorithms. 3) Existing multi-domain learning studies on neuroimaging data analysis.

Chapter 3 A Two-Stage Pipeline for Semi-Supervised Single Source Adaptation. This chapter aims to understand the feasibility of domain adaptation approaches to the small-sample, high-dimensional neuroimaging data analysis problems. For this purpose, a domain adaptation pipeline is proposed for whole brain fMRI decoding and evaluated on public fMRI datasets. This pipeline consists of two steps: feature adaptation and classifier adaptation. For implementation, transfer component analysis (Pan et al., 2011) and cross-domain SVM (Jiang et al., 2008) are employed at each step of the pipeline. This study adopted a workflow: learning, evaluation, and interpretation, which is also followed by the

studies in Chapter 5 and 6. Based on the experimental results and findings, two important research questions will be pointed out, where the multi-source domain adaptation will be the primary focus of Chapter 4 and 5.

Chapter 4 Dependence-Based Generalisation Theory. This chapter extends the statistical learning theory for the generalisation bounds of multi-domain learning. By leveraging the concept from multi-class classification, generalisation bounds for multi-source domain adaptation will be categorised as "one vs one" and "one vs rest" bounds. Analysis and derivations will be given to understand the relationships between these generalisation bounds, and a correlation between distribution divergence and statistical independence measurement. Then a dependence-based generalisation bound will be derived.

Chapter 5 Covariate-Independence Regularisation for Unsupervised Multi-Source Adaptation. This chapter aims to learn generalised patterns for multiple neuroimaging datasets. Public neuroimaging datasets with homogenous brain conditions from the OpenNeuro (Gorgolewski et al., 2017) repository are selected to construct a novel multi-source domain adaptation task for brain decoding. Moreover, following the theoretical analysis in the previous chapter, a framework for multi-source domain adaptation is proposed. Under this framework, there are two algorithms implemented by incorporating hinge and least squares loss. We study both algorithms on not only brain decoding but also textual sentiment classification and visual object recognition to investigate their efficacy across multiple applications.

Chapter 6 Covariate-Dependence Learning for Recognising Domain-Specific Patterns This chapter aims to learn domain-specific neuroimaging data. A neuroscience research problem, gender-related brain lateralisation, is formulated as a left / right hemisphere classification problem, and the objective is to learn a gender-specific classifier. Based on the theoretical results in Chapter 2 and 4, a framework will be proposed for learning domain-dependent models, and a variant of logistic regression will be derived under this framework. Effectiveness of the methods will be evaluated on the left / right classification tasks with the resting-state fMRI data from the HCP project.

Chapter 7 Conclusion. This chapter will summarise the results and findings of the above chapters and then point out the potential future research directions based on the studies involved in this thesis.

1.5 Contributions

This thesis includes theoretical and algorithmic contributions to the field of machine learning, and novel applications of neuroimaging data analysis. The **contributions** are summarised as following:

- **Feasibility**. A domain adaptation framework is proposed for whole brain fMRI decoding, which consists of two steps: feature adaptation, and classifier adaptation to reduce the generalisation error bound on target domain. Experimental results on OpenNeuro data confirm the feasibility of domain adaptation approach for functional neuroimaging analysis, with interpretable model provided.

- **Theory**. A statistical dependence based generalisation theory is derived for multi-domain learning. A proportional relationship between *Hilbert-Schmidt Independence Criterion (HSIC)* and *Maximum Mean Discrepancy (MMD)* is developed and results in a HSIC-based generalisation bound for multi-source domain adaptation.

- **Learning Frameworks**. The theoretical studies enable the formulation of two machine learning frameworks. One is *Covariate-Independence Regularisation (CoIR)* framework for multi-source domain adaptation that simultaneously minimises the empirical prediction risk on source domain and the dependence on domain covariates, to learn generalised models across domains. The other is *Covariate-Dependent Machine Learning (CoDeML)* framework that minimises the empirical prediction risk on target domain and maximises the dependence on domain covariates, to learn domain-specific models.

- **Algorithms**. Under the CoIR framework, a simplified HSIC is constructed and incorporated with the hinge and least square loss that can take unlabelled samples into account following the Manifold Regularization framework formulation (Belkin et al., 2006). This gives the CoIR_{SVM} and CoIR_{LS} algorithm. Under the CoDeML framework,

with the simplified HSIC above, a *Covariate-Dependent Logistic Regression (CoDeLR)* algorithm is derived by maximising the likelihood of sample classes and domain covariate dependence.

- **Novel domain adaptation tasks for brain decoding**. New learning tasks are constructed by identifying datasets with homogeneous brain conditions from public repositories. Experiments on these brain decoding tasks, together with the popular textual sentiment and visual recognition tasks, show the superior performance of CoIR over competing methods.

- **Novel domain-specific learning task**. The problem of gender-related brain lateralisation in neuroscience is formulated as a left / right hemisphere classification problem for CoDeML. Experiments on the HCP resting-state fMRI data show gender-specific models can be learnt by CoDeLR, compared to a standard logistic regression.

Chapter 2

Fundamentals of Multi-Domain Learning

In Chapter 1, some basic concepts of multi-domain learning have been introduced. This chapter will further provide more detailed background of this research problem. A machine learning model is called generalisable if it can assign labels or values to unseen samples correctly, which is also expected. This chapter will start with the theoretical support behind generalisability under the Probably Approximately Correct (PAC) framework, and then extends to recent theoretical studies of generalisation on multi-domain learning problems. Following the statistical theory, popular methods and applications will be presented, and this chapter will end with the recent multi-domain learning works on neuroimaging analysis.

2.1 Statistical Theory of Generalisation

Machine learning is a technique of learning from data. Suppose giving a sample dataset $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}, \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}, i \in [1, m]$, where $\mathcal{X}$ denotes the *input space*, i.e. the set of all possible instances or examples, $\mathcal{Y}$ denotes the output space, i.e. all possible target values or labels. For convenience, the chapter only consider binary classification problem, i.e. $|\mathcal{Y}| = 2$, and $\mathcal{Y} \in \{0, 1\}$ if not specified. A concept (or labelling function) $c : \mathcal{X} \to \mathcal{Y}$ is defined as a mapping from input space $\mathcal{X}$ to output space $\mathcal{Y}$. The task of learning is: selecting a hypothesis h from a set of possible concepts $\mathcal{H}$, which is called hypothesis set, to

approximate the target concept c.

Because the target concept is unknown, the key question is the feasibility of learning c with limited examples in S. The following Probably Approximately Correct framework will provide an attempt to answer this question and a guarantee from a theoretical perspective.

2.1.1 The PAC Learning Framework

The *Probably Approximately Correct (PAC)* is a theoretical framework developed from the "learnability" analysis of machine learning proposed by Valiant (1984). It helps to understand the number of training examples needed to obtain an approximate learnable concept. It can also provide generalisation bounds (theoretical guarantee) for the cost of computational learning and the guidance for algorithm design. This section will firstly introduce several essential definitions and notations for presenting the PAC learning framework, which will also be used throughout the thesis.

In the learning task defined above, the instances of $\mathcal{X}$ are assumed following an unknown distribution $\mathcal{D}$, and the samples in S are drawn independently from $\mathcal{D}$, i.e. independent and identically distributed (i.i.d). For a hypothesis h, the risk (or error) of disagreeing with a concept c is called generalisation risk, which is defined as follows:

Definition 2.1 (Generalisation risk)**.** *Given a hypothesis $h \in \mathcal{H}$, a target concept c: $\mathcal{X} \to [0, 1]$, the generalisation risk or error of h on a distribution $\mathcal{D}$ is*

$$\mathrm{R}(h, c) = \mathbb{P}_{\mathbf{x}\sim\mathcal{D}}\big[h(\mathbf{x}) \neq c(\mathbf{x})\big] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\big[|h(\mathbf{x}) - c(\mathbf{x})|\big]. \tag{2.1}$$

For convenience, the shorthand notation $\mathrm{R}(h) = \mathrm{R}(h, c)$ will be used in the rest of the thesis. By reusing the above definition, for an arbitrary pair of hypotheses $(h, h') \in \mathcal{H}^2$, we have

$$\mathrm{R}(h, h') = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\big[|h(\mathbf{x}) - h'(\mathbf{x})|\big]. \tag{2.2}$$

Due to the distribution $\mathcal{D}$ and target concept c are unknown, the generalisation risk cannot be directly estimated. However the *empirical risk (or error)* of a hypothesis $h \in \mathcal{H}$ can be estimated directly on sample set S, which is defined as follows:

Definition 2.2 (Empirical risk)**.** *Given a hypothesis $h \in \mathcal{H}$, a target concept c: $\mathcal{X} \to [0, 1]$, the empirical risk of h on a sample set $S = \{\mathbf{x}_i, y_i\}_{i=1}^m$ is*

$$\hat{\mathrm{R}}(h) = \frac{1}{n} \sum_{i=1}^{n} \big[h(\mathbf{x}_i) \neq c(\mathbf{x}_i)\big]. \tag{2.3}$$

The empirical risk (error) $\hat{\mathrm{R}}(h)$ obtained within the training sample set is also called the *in-sample error* $\mathrm{R}_{\text{in}}(h)$, and the generalisation risk (error) $\mathrm{R}(h)$ is also called the *out-of-sample error* $\mathrm{R}_{\text{out}}(h)$. By having the definition of generalisation and empirical risk, the PAC learning framework can be defined as follows:

Definition 2.3 (PAC-learning (Valiant, 1984))**.** *Let $\mathbf{X}$ be samples of size m drawn i.i.d from any distribution $\mathcal{D}$, a concept class $\mathcal{C}$ is PAC-learnable if exists an algorithm $\mathcal{A}$ and a polynomial function* $\mathrm{poly}(\cdot, \cdot, \cdot, \cdot)$ *so that for $\epsilon > 0$, $\delta \in (0, 1)$, and $\forall c \in \mathcal{C}$, the following holds for any $m \geq \mathrm{poly}(1/\epsilon, 1/\delta, \mathrm{size}(\mathbf{x}), \mathrm{size}(c))$:*

$$\mathbb{P}(\mathrm{R}(h_{\mathbf{X}}) \leq \epsilon) \geq 1 - \delta. \tag{2.4}$$

where the minimum of $m \geq \mathrm{poly}(1/\epsilon, 1/\delta, \mathrm{size}(\mathbf{x}), \mathrm{size}(c))$ that required by the learning algorithm $\mathcal{A}$ is called the sample complexity.

In the definition of PAC learnability, the confidence parameter δ can indicate the probability of approximated classifiers to satisfy the accuracy requirements, i.e. "probably", and the accuracy parameter ϵ determines how far the classifier can be from the target concept, i.e. "approximately correct" (Shalev-Shwartz & Ben-David, 2014). The complexity of hypothesis space $\mathcal{H}$ is another key point of PAC-learning. In general, the larger $\mathcal{H}$, the higher probability that a target concept $c \in \mathcal{H}$. However the higher complexity can also increase the difficulty of learning c. The following subsection will present the metrics for measuring complexity space.

2.1.2 VC-Dimension and Rademacher Complexity

The hypothesis set in real world machine learning tasks is usually infinite (e.g. hyper-planes in $\mathbb{R}^d$). In the following two popular metrics, VC-dimension and Rademacher complexity will

be introduced to prove that infinite hypothesis space can be measured and it is PAC-learnable. Before that, the definitions of *growth function* and *dichotomy* need to be introduced first:

Definition 2.4 (Growth function (Vapnik & Chervonenkis, 1971))**.** *Given a hypothesis set $\mathcal{H}$, $\forall m \in \mathbb{N}$, the growth function is*

$$\Pi_{\mathcal{H}}(m) = \max_{\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}\subseteq\mathcal{X}} \big|\{h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m)|h \in \mathcal{H}\}\big|. \tag{2.5}$$

Although the hypothesis set is infinite, the number of ways to label the sample set S is finite. For m samples in a binary classification problem, there are 2^m hypotheses in maximum. The growth function $\Pi_{\mathcal{H}}(m)$ is the maximum number of unique solutions to classify m sample points with hypotheses in $\mathcal{H}$. Each unique classification is called a *dichotomy*. When all possible dichotomies of a sample set S can be realised by the hypotheses in $\mathcal{H}$, the set S is said to be *shattered* by $\mathcal{H}$. By Hoeffding Inequality (Lemma A.1) and Definition 2.4, the following bound can be derived for generalisation risks:

Theorem 2.5 (Growth function generalisation bound (Vapnik & Chervonenkis, 1971))**.** *Given a hypothesis set $\mathcal{H}$, for $m \in \mathbb{N}$, $\epsilon \in (0,1)$, for any $h \in \mathcal{H}$*

$$\mathrm{R}(h) \leq \hat{\mathrm{R}}(h) + \sqrt{\frac{8}{m}\ln\frac{4\Pi_{\mathcal{H}}(2m)}{\delta}}. \tag{2.6}$$

Proof. See Appendix A. □

VC Generalisation Bound

Definition 2.6 (Vapnik–Chervonenkis (VC) dimension (Vapnik & Chervonenkis, 1971; Vapnik, 2006))**.** *The VC-dimension of a hypothesis set $\mathcal{H}$, is the largest value of m for a sample set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ can be shattered by $\mathcal{H}$, where $y_i \in \{-1, 1\}$:*

$$\mathrm{VC}(\mathcal{H}) = \max\{n : \forall \mathbf{x}_i, y_i \in \{(\mathbf{x}_i, y_i)\}_{i=1}^m, \exists h \in \mathcal{H}, \textit{so that } h(\mathbf{x}_i) = y_i\} \tag{2.7}$$

Theorem 2.7 (VC generalisation bound (Vapnik & Chervonenkis, 1971; Vapnik, 2006))**.** *Given a hypothesis set $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$, $\mathcal{Y} = \{-1, +1\}$, with VC-dimension $\mathrm{VC}(\mathcal{H})$, then for any*

$\delta \in (0, 1]$, *with probability of at least* $1 - \delta$, $\forall h \in \mathcal{H}$, *the following holds*

$$\hat{\mathrm{R}}(h) \leq \hat{\mathrm{R}}(h) + \sqrt{\frac{8}{m}\big(\mathrm{VC}(\mathcal{H}) \ln \frac{2em}{\mathrm{VC}(\mathcal{H})} + \ln \frac{4}{\delta}\big)}. \tag{2.8}$$

The form of the VC generalisation can be written as

$$\mathrm{R}(h) \leq \hat{\mathrm{R}}(h) + \mathrm{O}\Big(\sqrt{\frac{\ln(m/\mathrm{VC}(\mathcal{H}))}{(m/\mathrm{VC}(\mathcal{H}))}}\Big), \tag{2.9}$$

where $\mathrm{O}(\cdot)$ denotes the computational complexity. Equation 2.9 highlights the the importance of the ratio $\mathrm{VC}(\mathcal{H})/m$, which indicates that with the increasing of training instances and lower complexity of hypothesis set, the generalisation risk will get close to the empirical risk. It is also the reason that many machine learning algorithms optimise empirical prediction loss and regularisation on model parameters as objectives.

Rademacher Complexity Generalisation Bound

The data distribution $\mathcal{D}$ is not considered in VC-dimension and therefore the VC generalisation bound is distribution-independent. By contrast, *Rademacher Complexity* (Koltchinskii & Panchenko, 2000; Koltchinskii, 2001; Bartlett et al., 2002) is another approach for measuring the complexity of hypothesis space, which can take empirical data distribution into consideration.

Definition 2.8 (Empirical Rademacher complexity (Koltchinskii & Panchenko, 2000; Koltchinskii, 2001))**.** *Given a hypothesis set* $\mathcal{H}$*:* $\mathcal{X} \to \mathcal{Y}$*, and* $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ *a fixed sample set of size* m *from* $\mathcal{X}$*, the empirical Rademacher complexity of* $\mathcal{H}$ *w.r.t* $\mathbf{X}$ *is*

$$\hat{\mathfrak{R}}_{\mathbf{X}}(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h(\mathbf{x}_i)\Big], \tag{2.10}$$

where $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1}^n$ *is a set of random variables defined as*

$$\sigma = \begin{cases} 1, \textit{with probability } \frac{1}{2} \\ -1, \textit{otherwise}, \end{cases}$$

which is called the Radmecher variable.

The empirical Rademcher complexity measures the correlation between of a hypothesis set ($\mathcal{H}$) and random noise presented in the sample set S. By sampling m samples from distribution $\mathcal{D}$ i.i.d, resulting the expectation as:

Definition 2.9 (Rademacher complexity (Koltchinskii & Panchenko, 2000; Koltchinskii, 2001))**.** *For $m \geq 1$, the Rademacher complexity of a hypothesis set $\mathcal{H}$ is the expectation of the empirical Rademacher complexity over finite sample $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^m$ of size m drawn from the distribution $\mathcal{D}$, i.e.*

$$\Re_m(\mathcal{H}) = \mathbb{E}_{\mathbf{X}\sim\mathcal{D}^m}[\hat{\Re}(\mathcal{H})]. \tag{2.11}$$

Song (2008) introduced a relationship between Rademacher complexity and kernel representations in a reproducing kernel Hilbert space (RKHS). For a RKHS $\mathcal{H}$, $\forall h \in \mathcal{H}$, where $\|h\|_{\mathcal{H}} \leq 1$:

$$\begin{aligned}
\Re_m(\mathcal{H}) &= \mathbb{E}_{\mathbf{X}}\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i h(\mathbf{x}_i)\Big] \\
&= \mathbb{E}_{\mathbf{X}}\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i h(\mathbf{x}_i)\Big] \\
&= \mathbb{E}_{\mathbf{X}}\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \Big\langle h, \frac{1}{m}\sum_{i=1}^m \sigma_i k(\mathbf{x}_i)\Big\rangle\Big] \\
&= \frac{1}{m}\mathbb{E}_{\mathbf{X}}\mathbb{E}_{\boldsymbol{\sigma}}\Big[\Big\|\sum_{i=1}^m \sigma_i k(\mathbf{x}_i)\Big\|_{\mathcal{H}}\Big] \\
&\leq \frac{1}{m}\mathbb{E}_{\mathbf{X}}\Big[\Big(\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sum_{i,j=1}^m \sigma_i\sigma_j k(\mathbf{x}_i,\mathbf{x}_j)\Big]\Big)\Big] \\
&= \frac{1}{m}\mathbb{E}_{\mathbf{X}}\Big[\sqrt{\mathrm{tr}(\mathbf{K})}\Big],
\end{aligned} \tag{2.12}$$

where $k(\cdot)$ denotes a kernel function, and $\mathbf{K} \in \mathbb{R}^{m\times m}$ denotes a kernel matrix.

Based on Definition 2.9, the following theorem presents the Rademacher complexity bound

Theorem 2.10 (Rademacher complexity bound (Koltchinskii, 2001))**.** *Let $\mathcal{H}$ be a hypothesis set: $\mathcal{X} \to \mathcal{Y}$, and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^m$ a finite sample set of size m drawn from distribution. Then*

for any $\delta \in (0, 1]$*, with probability of at least* $1 - \delta$*,* $\forall h \in \mathcal{H}$*, the following holds*

$$\mathrm{R}(h) \leq \hat{\mathrm{R}}(h) + \Re_m(\mathcal{H}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}, \tag{2.13}$$

and

$$\mathrm{R}(h) \leq \hat{\mathrm{R}}(h) + \hat{\Re}_{\mathbf{X}}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{1}{\delta}}{2m}}. \tag{2.14}$$

2.2 Theory of Learning from Multiple Domains

Section 2.1 summarises the popular theoretical analysis framework for machine learning algorithms, where the data samples are assumed to come from the same distribution (domain). However, for the multi-domain problem mentioned in Chapter 1, the generalisation bounds introduced previously are not applicable. This section will present the theoretical guarantees for learning from multiple domains. The learning problem of single-source domain adaptation will be defined mathematically first and followed by an introduction of the generalisation bounds for domain adaptation.

2.2.1 Problem Definition of Domain Adaptation

Given a source domain labelled sample set $S = \{(\mathbf{x}_1^s, y_1^s), (\mathbf{x}_2^s, y_2^s), \ldots, (\mathbf{x}_{m_s}^s, y_{m_s}^s)\}$ and an unlabelled target set $T = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots \mathbf{x}_{m_t}^t\}$, the objective is learning a classifier (hypothesis) h for the target domain, using the knowledge from S and T (Pan & Yang, 2010). Behind the learning problem, there are **three main assumptions** (Pan & Yang, 2010; Redko et al., 2020):

1. **Covariate shift**. Denoting $\mathbb{P}_s(\mathbf{x})$ and $\mathbb{P}_s(y|\mathbf{x})$ as the marginal and conditional distribution of a source domain, respectively, using the same way of notation for a target domain, it is assumed that $\mathbb{P}_s(\mathbf{x}) \neq \mathbb{P}_t(\mathbf{x})$, $\mathbb{P}_s(y|\mathbf{x}) = \mathbb{P}_t(y|\mathbf{x})$.

2. **Domain marginal distribution similarity**. This assumption assumes that the feasibility / learnability of domain adaptation can be assessed by the divergence (or similarity) between the marginal distributions of source and target. This is a straightforward

assumption for deriving the generalisation bounds presented later.

3. **Risk of ideal joint hypothesis**. The assumption assumes an existing ideal hypothesis, which is low-risk for both domains. If the combined risk of the ideal joint hypothesis is high, it is not possible to expect to learn a good classifier by minimising the error on the source data set. The ideal joint hypothesis is defined as follows:

   Definition 2.11 ((Ben-David et al., 2010)). *Given a hypothesis set $\mathcal{H}$, the ideal joint hypothesis is the hypothesis which minimises the combined error*

$$h^* = \arg\min_{h \in \mathcal{H}} \big(\mathrm{R}_s(h) + \mathrm{R}_t(h)\big). \tag{2.15}$$

   The combined risk of the ideal hypothesis is denoted as $\lambda^* = \mathrm{R}_s(h^*) + \mathrm{R}_t(h^*)$.

2.2.2 Domain Divergence Metric and Generalisation Bounds

As mentioned in the above assumptions, distribution divergence between source and target is an important criteria to assess the learnability of a domain adaptation problem. In this subsection, three popular divergence metric, *$\mathcal{H}$-divergence* (Ben-David et al., 2007), *symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$* (Ben-David et al., 2010), and *maximum mean discrepancy (MMD)* (Borgwardt et al., 2006), will be introduced and then presenting the generalisation bounds based on these metrics.

Definition 2.12 ($\mathcal{H}$-divergence (Ben-David et al., 2007)). *Let $\mathcal{H}$ be a hypothesis set, $\mathbf{X}^t$ and $\mathbf{X}^s$ are sample sets drawn from distributions $\mathcal{D}^s$ (source) and $\mathcal{D}^t$ (target), respectively. Then for $h \in \mathcal{H}$, the $\mathcal{H}$-divergence between $\mathcal{D}^s$ and $\mathcal{D}^t$ is*

$$\hat{d}_{\mathcal{H}}(\mathcal{D}^t, \mathcal{D}^s) = 2 \sup_{h \in \mathcal{H}} |\mathbb{P}_{\mathbf{x}^t \sim \mathcal{D}^t}[h(\mathbf{x}^t) = 1] - \mathbb{P}_{\mathbf{x}^s \sim \mathcal{D}^s}[h(\mathbf{x}^s) = 1]|. \tag{2.16}$$

Here an example is given to explain $\mathcal{H}$-divergence, let $\mathbf{X}^s = [\mathbf{x}_1, \ldots \mathbf{x}_{10}]$, $\mathbf{X}^t = [\mathbf{x}_{11}, \ldots \mathbf{x}_{20}]$, $\mathcal{H} = \{h_1, h_2\}$, where

- $h_1(\mathbf{x}_i) = 1, i = 1 \ldots 5, 11 \ldots 15$, $h_1(\mathbf{x}_i) = 0, i = 6 \ldots 10, 16 \ldots 20$, and
- $h_2(\mathbf{x}_i) = 1, i = 1 \ldots 10$, $h_2(\mathbf{x}_i) = 0, i = 11 \ldots 20$,

and therefore

- $\mathbb{P}_{\mathbf{x}^t}[h_1(\mathbf{x}_i^t) = 1] = 0.5$, $\mathbb{P}_{\mathbf{x}^s}[h_1(\mathbf{x}_i^s) = 1] = 0.5$,
- $\mathbb{P}_{\mathbf{x}^t}[h_2(\mathbf{x}_i^t) = 1] = 1$, $\mathbb{P}_{\mathbf{x}^s}[h_2(\mathbf{x}_i^s) = 1] = 0$.

So in this case $\hat{d}_{\mathcal{H}}(\mathcal{D}_t, \mathcal{D}_s) = 1$.

$\mathcal{H}$-divergence can measure the divergence between two distributions. However, it cannot be linked to a hypothesis set for using in a generalisation bound, and therefore Ben-David et al. (2010) further proposed the *symmetric difference hypothesis space* $\mathcal{H}\Delta\mathcal{H}$, which is defined as follows

Definition 2.13 (Symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$ (Ben-David et al., 2010))**.** *For a hypothesis space $\mathcal{H}$, the symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H}$ is the set of hypothesis*

$$\mathcal{H}\Delta\mathcal{H} = \{h(\mathbf{X}) \oplus h'(\mathbf{X}) | h, h' \in \mathcal{H}\},$$

where $\oplus$ is the XOR operation.

In terms of the symmetric difference hypothesis space, a generalisation bound on the target risk is derived as follows (Ben-David et al., 2010):

Theorem 2.14 (Single-source domain adaptation generalisation bound (Ben-David et al., 2010))**.** *Let $\mathcal{H}$ be a hypothesis space of VC dimension* $\mathrm{VC}(\mathcal{H})$*, and $\mathbf{X}^s$, $\mathbf{X}^t$ are sample sets of size m drawn from $\mathcal{D}^s$ and $\mathcal{D}^t$, respectively, then for any $\delta \in (0, 1)$, with the probability at least $1 - \delta$, $\forall h \in \mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \mathrm{R}_s(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}^t, \mathbf{X}^s) + \sqrt{\frac{1}{m}\big(2\mathrm{VC}(\mathcal{H}) \ln 2m + \ln \frac{2}{\delta}\big)} + \lambda^*, \tag{2.17}$$

where λ^* is the combined risk of the ideal hypothesis as defined in Definition 2.11.

Maximum mean discrepancy (MMD) is one of the integral probability metrics. It is a popular metric of distribution divergence used by domain adaptation algorithms. It can be computed effectively from finite available samples, which is defined as follows

Definition 2.15 (Maximum mean discrepancy (MMD) (Borgwardt et al., 2006)). *For $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leq 1\}$, where $\mathcal{H}$ is a reproducing kernel Hilbert space, the maximum mean discrepancy between two distributions $\mathcal{X}$ and $\mathcal{Y}$ is*

$$\hat{d}_{\text{MMD}}(\mathcal{X}, \mathcal{Y}) = \sup_{f \in \mathcal{F}} \Big| \int f \mathrm{d}(\mathcal{X} - \mathcal{Y}) \Big|. \tag{2.18}$$

The empirical MMD between two sample sets $\mathbf{X}$ and $\mathbf{Y}$ can be computed via

$$\begin{aligned} \text{MMD}[\mathcal{F}, \mathbf{X}, \mathbf{Y}] &= \sup_{f \in \mathcal{F}} \Big(\frac{1}{m_x} \sum_{i=1}^{m_x} f(\mathbf{x}_i) - \frac{1}{m_y} \sum_{i=1}^{m_y} f(\mathbf{y}_i) \Big) \\ &= \Big[\frac{1}{m_x^2} \sum_{i,j=1}^{m_x} k(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{m_x m_y} \sum_{i,j=1}^{m_x, m_y} k(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m_y^2} \sum_{i,j=1}^{m_y} k(\mathbf{y}_i, \mathbf{y}_j) \Big], \end{aligned} \tag{2.19}$$

where $\mathcal{H}$ denotes a reproducing kernel Hilbert space (RKHS) (Berlinet & Thomas-Agnan, 2011), $k(\cdot, \cdot)$ denotes a kernel function, such as linear, Gaussian, and polynomial, $\mathbf{x}_i$ and $\mathbf{y}_j$ are the ith and jth sample of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Unlike the $\mathcal{H}\Delta\mathcal{H}$-based generalisation bound, which was derived under the framework of VC-dimension, Redko et al. (2020) derived the following MMD-based theorem under the framework of Rademachmer complexity:

Theorem 2.16 ((Redko et al., 2020)). *Let $\mathcal{H}$ be a hypothesis space, $\mathbf{X}^s$, $\mathbf{X}^t$ are samples with size m drawn from $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively, then for any $\delta \in (0, 1)$ with probability of at least $1 - \delta$, $\forall h \in \mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \mathrm{R}_s(h) + \hat{d}_{\text{MMD}}(\mathbf{X}^t, \mathbf{X}^s) + \frac{1}{n} \big(\mathbb{E}_{\mathbf{X}^t} \big[\sqrt{\text{tr}(\mathbf{K}_t)} \big] + \mathbb{E}_{\mathbf{X}^s} \big[\sqrt{\text{tr}(\mathbf{K}_s)} \big] \big) + 2 \sqrt{\frac{\ln \frac{2}{\delta}}{2m}} + \lambda^*. \tag{2.20}$$

Generalisation Bounds for Multi-Source Domains

Compared to learning from a single source domain, learning from multiple source domains is a practical problem in real world tasks. Ben-David et al. (2010) proved the following two $\mathcal{H}\Delta\mathcal{H}$-based theorems for multi-source domain adaptation. Let $\mathbf{X}^s$ be a sample set of samples drawn from J distributions, the first generalisation bound can be obtained by applying Theorem 2.14 J times and can be summarised as "one vs target" bound:

Theorem 2.17 (One vs target (Ben-David et al., 2010)). *Let $\mathcal{H}$ be a hypothesis space of VC dimension $\mathrm{VC}(\mathcal{H})$. For each $j \in \{1, \ldots, J\}$, let $\mathbf{X}_j$ be labelled sample of size $\beta_j m$ drawn from distribution $\mathcal{D}_j$ with weight α_j, then for any $\delta \in (0, 1)$, with the probability at least $1 - \delta$, for every $h \in \mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \mathrm{R}_s(h) + \sum_{j=1}^{J} \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}^t, \mathbf{X}_j^s) + \lambda_j\big) + \mathrm{O}\sqrt{\sum_{j=1}^{J} \big(\frac{\alpha_j^2}{\beta_j}\big)\frac{1}{m}\big(\mathrm{VC}(\mathcal{H}) \ln m + \ln \frac{1}{\delta}\big)}, \quad (2.21)$$

where $\lambda_j = \min_{h \in \mathcal{H}}\{\mathrm{R}_j(h) + \mathrm{R}_t(h)\}$.

Ben-David et al. (2010) also proved a tighter generalisation bound in theory by considering the mixture of source domain sets as a single source, i.e. "*source combine*" strategy used in many domain adaptation studies as baselines, such as (Zhu et al., 2019; Peng et al., 2019), which gives the following theorem:

Theorem 2.18 (Source combine (Ben-David et al., 2010)). *Let $\mathcal{H}$ be a hypothesis space of VC dimension $\mathrm{VC}(\mathcal{H})$. For each $j \in \{1, \ldots, J\}$, let $\mathbf{X}_j$ be labelled sample of size $\beta_j m$ drawn from distribution $\mathcal{D}_j$ with weight α_j that constitutes the mixture $\mathbf{X}^s$, then for any $\delta \in (0, 1)$, with the probability at least $1 - \delta$, for every $h \in \mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \mathrm{R}_s(h) + \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}^s, \mathbf{X}^t) + \lambda_{\boldsymbol{\alpha}} + \mathrm{O}\sqrt{\sum_{j=1}^{J} \big(\frac{\alpha_j^2}{\beta_j}\big)\frac{1}{m}\big(\mathrm{VC}(\mathcal{H}) \ln m + \ln \frac{1}{\delta}\big)}, \quad (2.22)$$

where $\lambda_{\boldsymbol{\alpha}} = \min_{h \in \mathcal{H}}\{\sum_{j=1}^{J} \alpha_j \mathrm{R}_j(h) + \mathrm{R}_t(h)\}$.

The generalisation bounds of multi-domain learning, including Eq. (2.21), (2.22), and single source bounds such as Eq. (2.17) can be summarsed as the following equation:

$$\mathrm{R}_t(h) \leq \hat{\mathrm{R}}_s(h) + \text{Divergence} + \text{Complexity} + \text{Constant}, \quad (2.23)$$

where the major difference is the divergence term. To better understand the two multi-source generalisation bounds, assume there are three domains: A, B, and C, and A is the target domain. In terms of tightness, the relationship between the divergences in Eq. (2.21) and

(2.22) can be summarised as the following inequality:

$$\underbrace{|\mathrm{A}-\mathrm{BC}|}_{\text{“Source Combine”}} \leq \underbrace{|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|}_{\text{“One vs Target”}},$$

where $|\mathrm{A}-\mathrm{BC}|$ is the source combine divergence and $|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|$ is the one vs target divergence.

2.3 Multi-Domain Learning Methods

By Theorem 2.14 $\sim$ 2.18, the generalisation error are determined by an additional divergence between source and target domains comparing to the standard generalisation bounds, as summarised in Eq. (2.23). Therefore, the basic idea behind domain adaptation in practice is minimising the domain divergence. Popular domain adaptation methods will be introduced in the following three categories: feature mapping learning, cross-domain classifier learning, and deep neural networks.

2.3.1 Feature Mapping Learning

One of the most popular approaches of domain adaptation is to minimise the distribution mismatch between a source and a target, typically measured by the *maximum mean discrepancy* (MMD) criterion (Borgwardt et al., 2006), via learning one (or two) mapping(s) of the input data to a subspace. Pan et al. (2011) proposed *Transfer Component Analysis* (TCA) to learn such a mapping by minimising the MMD of marginal distribution mismatch and maximising the captured variances. Pan et al. (2011) also proposed *Semi-Supervised TCA* (SSTCA) for extracting more discriminate features by maximising HSIC on the data labels. Gong et al. (2016) proposed *conditional transferable components* (CTC) to minimise the conditional distribution mismatch. Long et al. (2013b) extended TCA to *Joint Distribution Adaptation* (JDA) to minimise both marginal and conditional distribution mismatch. We can summarise these methods in a general formula

$$\min_{\phi} d(\phi(\mathbf{X}^s), \phi(\mathbf{X}^t)) - \mathrm{Var}(\phi(\mathbf{X})), \tag{2.24}$$

Table 2.1: Six feature mapping learning methods as in Eq. (2.24). # of ϕ denotes the number of learnt feature mappings.

Method	**Objective to Minimise**	**Objective to Maximise**	# of ϕ
TCA (Pan et al., 2011)	MMD (marginal)	Captured variance	One
CTC (Gong et al., 2016)	MMD (conditional)	Target label dependence	One
JDA (Long et al., 2013b)	MMD (marginal + conditional)	Captured variance	One
VDA (Tahmoresnezhad & Hashemi, 2017)	∘ MMD (marginal + conditional) ∘ Within class covariance	Captured variance	One
BDA (Wang et al., 2018)	MMD (λ marginal + $(1-\lambda)$ conditional)	Captured variance	One
JGSA (Zhang et al., 2017)	∘ MMD (marginal + conditional) ∘ λ Subspace distance ∘ β Within class covariance	∘ μ Target covariance ∘ β Between class covariance	Two

where $d(\cdot,\cdot)$ denotes a function of domain divergence measurment, ϕ is a feature mapping, $\mathbf{X}^s$ and $\mathbf{X}^t$ are source and target domain data, respectively, $\mathcal{D}(\phi(\mathbf{X}^s),\phi(\mathbf{X}^t))$ is the distribution mismatch (MMD) between source and target domains, $\mathbf{X}$ is the combined source and target data, $\text{Var}(\cdot)$ denotes the total captured variance. A regularisation term on the feature mapping matrix is typically incorporated. SSTCA has a label dependence objective $-\mu\rho_h(\phi(\mathbf{X}),\mathbf{Y})$ added to Eq. (2.24), where μ is a hyper-parameter, and $\mathbf{Y}$ is a label matrix (e.g. $y_{i,j}=1$ if $\mathbf{x}_i$ belongs to the jth class; $y_{i,j}=0$ otherwise).

JDA has several extensions. For example, *visual domain adaptation* (VDA) and (Tahmoresnezhad & Hashemi, 2017) *Joint Geometrical and Statistical Alignment* (JGSA) (Zhang et al., 2017) incorporate JDA with Fisher's linear discriminant analysis (LDA), and *balanced distribution adaptation* (BDA) (Wang et al., 2017) introduces a trade-off between marginal and conditional distribution mismatch.

Domain Dependence Minimisation

Yan et al. (2018) proposed the method *Maximum independence domain adaptation* (MIDA)

using a new approach. It obtains cross-domain features by learning a mapping to minimise the dependence on auxiliary domain information. In contrast, such domain information is not directly modeled in the distribution mismatch minimisation mapping or domain-invariant classifier learning approaches. We summarise the objective of MIDA as

$$\min_{\phi} \rho_h(\phi(\mathbf{X}), \mathbf{C}) - \mathrm{Var}(\phi(\mathbf{X})), \tag{2.25}$$

where $\mathbf{C}$ encodes the auxiliary domain information (domain covariates). Similar to TCA, MIDA also has a semi-supervised version (SMIDA) that maximises label dependence to Eq. (2.25):

$$\min_{\phi} \rho_h(\phi(\mathbf{X}), \mathbf{C}) - \mathrm{Var}(\phi(\mathbf{X})) - \mu\rho_h(\phi(\mathbf{X}). \tag{2.26}$$

2.3.2 Cross-Domain Classifier Learning

Classifier Adaptation

Model/Classifier adaptation is another domain adaptation approach that aims at training a model for target domain using the knowledge, such as coefficients or parameters, from a pre-trained model. The key difference between domain invariant classifier and classifier adaptation is: the input for domain invariant classifier is data from source and target domain, while the input for classifier adaptation is a pre-trained (on source) model and target domain data.

Yang et al. (2007) proposed adaptive SVM (ASVM) to fit an pre-trained SVM for target domain data by adding a "delta function" $\Delta f(\mathbf{x}) = \mathbf{w}^\top \Phi(\mathbf{x})$. The overall obejctive of ASVM is given by

$$f(x) = \mathbf{w}^\top \Phi(\mathbf{x}) + f^s(\mathbf{x}) \tag{2.27}$$

Where $\Phi(\mathbf{x})$ is the feature vector, $\mathbf{w}$ is the weight vector, $f^s()$ is a classifier trained on the source data. The objective function of ASVM is same as the SVM except the constrains,

Table 2.2: The four domain-invariant classifier learning methods as in Eq. (2.29): *Semi-supervised kernel matching domain adaptation* (SSMDA) (Xiao & Guo, 2015), *Selective Transfer Machine* (STM) (Chu et al., 2017), *Distribution Matching Machine (DMM)* (Cao et al., 2018), and *Manifold Embedded Distribution Alignment* (MEDA) (Wang et al., 2018).

Method	$\mathcal{L}(f(\phi(\mathbf{X}^l),\mathbf{y}))$	$d(\phi(\mathbf{X}^s),\phi(\mathbf{X}^t))$	$\mathcal{M}(\phi(\mathbf{X}^s),\phi(\mathbf{X}^t))$
SSMDA	Square loss	$-\rho_h(\phi(\mathbf{X}^s),\phi(\mathbf{X}^t))$	Graph Laplacian
STM	Hinge loss	MMD (marginal)	$\times$
DMM	Hinge loss	MMD (marginal)	$\times$
MEDA	Square loss	MMD (λ marginal + $(1-\lambda)$ conditional)	Graph Laplacian

given by

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \\ &\text{s.t. } \xi_i \geq 0, y_i f^s(\mathbf{x}_i) + y_i \mathbf{w}^\top \phi(\mathbf{x}_i) \geq 1-\xi_i, \forall(\mathbf{x}_i,y_i) \in \mathbf{D}_l^t. \end{aligned} \tag{2.28}$$

where $\mathbf{D}_l^t$ denotes the labelled data of target domain, N is the number of samples of $\mathbf{D}_l^t$. C is a parameter which controls the extent that how large $f(\mathbf{x})$ is affected by $f^s(\mathbf{x})$. $f(\mathbf{x})$ would be highly influenced by $f^s(\mathbf{x})$ when C is small. While the effect of $f^s(\mathbf{x})$ would be small if C is large.

On the basis of ASVM, Jiang et al. (2008) proposed CDSVM, which leverages the source support vectors learnt by a standard SVM on source domain samples to find a better decision boundary for target samples. It re-weights each source support vector according to its average distance to the target (training) feature vectors, and then the target classifier will be trained with the target training samples and re-weighted source support vectors.

Domain-Invariant Classifier

Another approach learns a classifier by minimising the prediction error and distribution mismatch jointly. Long et al. (2013a) proposed a general framework, *Adaptation Regularisation based Transfer Learning* (ARTL). The formula of ARTL is

$$\min_{f,\phi} \mathcal{L}(f(\phi(\mathbf{X}^s),\mathbf{y}^s)) + \lambda d_{\text{MMD}}(\phi(\mathbf{X}^s),\phi(\mathbf{X}^t)) + \gamma\mathcal{M}(\phi(\mathbf{X}^s),\phi(\mathbf{X}^t)), \tag{2.29}$$

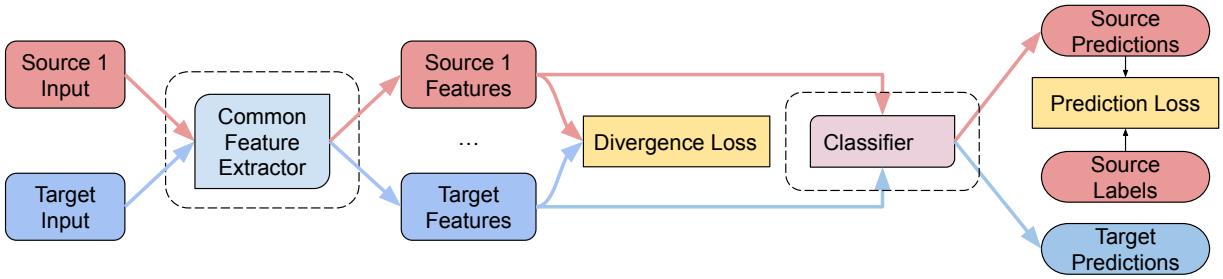


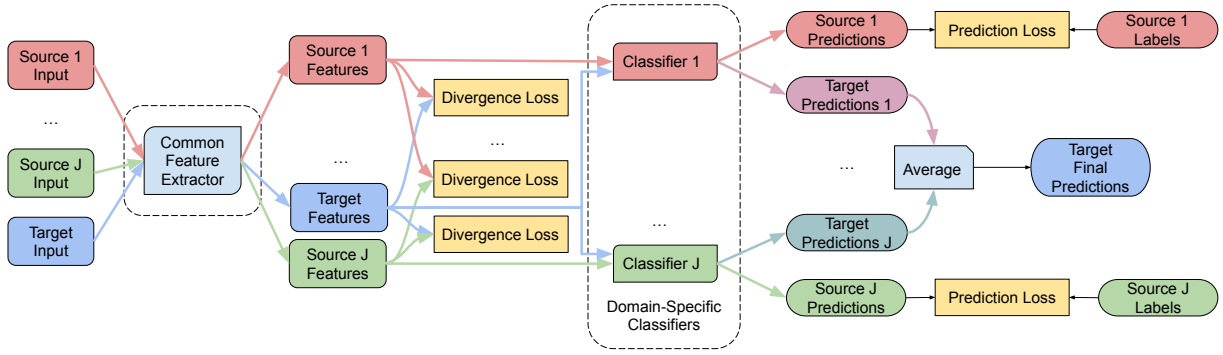
Figure 2.1: Illustration of general single-source domain adaptation networks. Neural network architectures (trainable parameters) are circled by dashed rounded rectangles.

where $f(\cdot)$ is a decision function of a classifier, $\mathcal{L}(f(\phi(\mathbf{X}^s), \mathbf{y}^s))$ denotes a loss function that measures the prediction error of labelled source domain samples, $\mathcal{M}(\phi(\mathbf{X}^s), \phi(\mathbf{X}^t))$ denotes the manifold regularisation (Belkin et al., 2006), $\mathbf{X}^l$ denotes labelled data, which can be source data only or source data plus target data according to different settings, and $\mathbf{y}$ is a data label vector. Table 2.2 summarises four such recent methods.

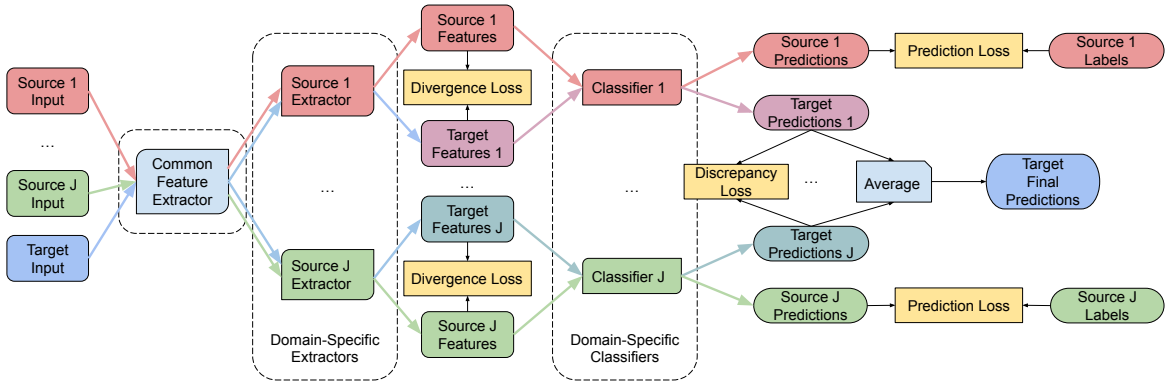
2.3.3 Domain Adaptation via Deep Neural Networks

The advances of deep learning on computer vision (CV) tasks also promote deep domain adaptation approaches. The simplest application is using the neural network trained large datasets (e.g. ImageNet) and "fine-tune" the last (or last few) layer(s) on the target data, where fixed layers are considered as a feature extractor. There are also more advanced deep DA networks trained end-to-end, i.e. learning feature embedding and classification networks simultaneously. Pre-trained convolutional neural networks (CNNs) are usually leveraged as backbones. Domain distribution divergence loss, such as adversarial loss in Ganin et al. (2016) and Maximum Mean Discrepancy (MMD) in Long et al. (2019) are used to regularise and fine-tune the parameters of embedding networks, and therefore general representations for different domains can be extracted and a generalised model can be learnt for target domains. Figure 2.1 demonstrates the architecture of these approaches.

While the single-source DA problem has been widely explored, multiple sources is a more practical scenario in applications. Recent research has shown the advantages of modelling multi-source domains over neglecting the differences between the multiple sources (i.e. applying single-source DA approaches to multi-source DA problems by viewing multiple source



(a) Moment Matching for Multi-Source Domain Adaptation (M^3SDA). This figure is reproduced according to descriptions in (Peng et al., 2019).



(b) Multiple Feature Spaces Adaptation Network (MFSAN). This figure is reproduced according to descriptions in (Zhu et al., 2019).

Figure 2.2: Two examples of deep neural networks for multi-source domain adaptation. Neural network architectures (trainable parameters) are circled by dashed rounded rectangles.

domains as a single one). For example, Peng et al. (2019) proposed *Moment Matching for Multi-Source Domain Adaptation (M^3SDA)* to learn domain generic features by minimising the distribution divergence between arbitrary pair of domains, as depicted in Fig. 2.2a. Additionally, there is a classifier specifically for each domain and final prediction is the ensemble of predictions given by these domain-specific classifiers. Zhu et al. (2019) proposed a more complicated deep neural network *Multiple Feature Spaces Adaptation Network (MFSAN)* for multi-source visual domain adaptation, where an additional domain-specific feature extractor is adopted for each domain. Figure 2.2b shows the architecture of this neural network.

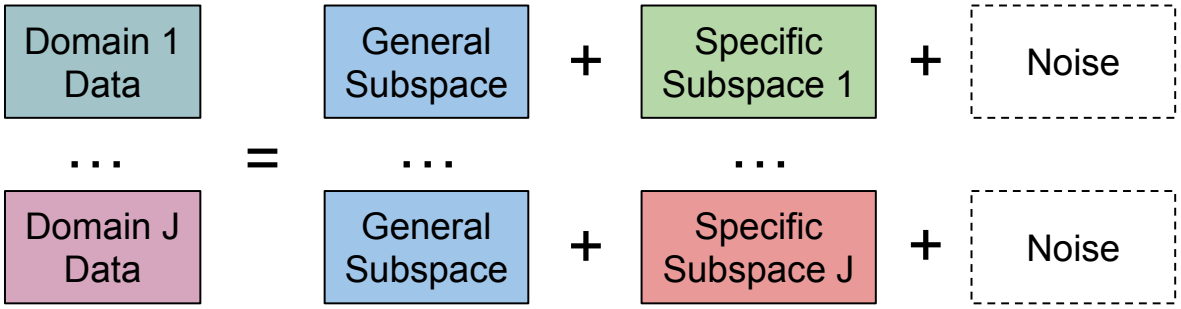


Figure 2.3: Assumptions of general and specific feature separation approaches, where the data of each domain is a mixture of general and specific features plus noise. The first two steps of these approaches are de-noising and extracting general features. Then domain specific-features can be obtained by subtracting general (reconstructed) features from de-noised data.

2.3.4 General and Specific Feature Separation

Similar to the domain adaptation methods, there is another branch of approaches (Lock et al., 2013; Zhou et al., 2015) also aim at extracting general features across different domains, while learn specific features for each domain at the mean time, i.e. general and specific feature separation. The problem is formulated as

$$\begin{aligned} &\sum_{n} \min_{\bar{\mathbf{A}},\breve{\mathbf{A}}_n} \|\mathbf{Y}_n - \bar{\mathbf{A}}\bar{\mathbf{B}}_n^\top - \breve{\mathbf{A}}_n\breve{\mathbf{B}}_n^\top\|, \\ &\text{s.t. } \bar{\mathbf{A}}\bar{\mathbf{A}}^\top = \mathbf{I}, \breve{\mathbf{A}}_n\breve{\mathbf{A}}_n^\top = \mathbf{I}, \bar{\mathbf{A}}\breve{\mathbf{A}}_n^\top = 0, \end{aligned} \tag{2.30}$$

where $\mathbf{Y}_n$ denotes the nth block of data, $\bar{\mathbf{A}}$, $\breve{\mathbf{A}}_n$ are the common basis/components for all data blocks/groups and specific basis/components for data block n, respectively, and $\bar{\mathbf{B}}_n$ and $\breve{\mathbf{B}}_n$ are the matrix of mixing coefficients for common and specific basis, respectively.

Here it is assumed that general and specific features are mixed by adding together, as shown in Figure 2.3. The core to solve Eq. (2.30) is de-noising and learning the common basis $\bar{\mathbf{A}}$ and then we can easily obtain individual features via subtraction.

2.3.5 Applications in Neuroimaging Analysis

Recently, domain adaptation techniques have been applied to studying human brain states. For example, Zhang et al. (2018a) proposed two approaches by making use of shared subjects between target and source fMRI datasets. They employed two factorisation models Varo-

quaux et al. (2011); Chen et al. (2015) to learn subject-specific bases, which are assumed to be invariant across datasets. Using a source dataset with shared subjects can help the models learn better subject-specific bases, and therefore improve the prediction accuracy. However, the approaches are not applicable when no subjects are shared between datasets, which is common for multi-site data sharing projects. Mensch et al. (2017) take a multi-task learning approach to use *resting-state data* from the Human Connectome Project (Van Essen et al., 2012) to learn a general representation via matrix decomposition and then jointly optimize multiple *heterogeneous* task-based fMRI classification tasks. Deep models such as autoencoder (Velioglu & Vural, 2017; Li et al., 2018) and AlexNet (Zhang et al., 2019) pre-trained on generic source data have also been used to represent the target fMRI data for classification. However, the source data used by these existing fMRI DA studies are independent to the target classification task.

There is another related multi-task learning (MTL) approach for neural decoding. Rao et al. (2013); Cox & Rogers (2021) proposed sparse overlapping sets lasso (SOS Lasso) for fMRI, with an MTL approach. MTL is a branch of domain adaptation, which aims at improving the performance of all tasks considered and does not differentiate source and target domains. In the context of SOS Lasso, the multiple "tasks" are datasets associated with each of several *participants* of *the same* experiment. The tasks are related as in multitask group lasso (Yuan & Lin, 2006), but groups can be overlapped with one another, and features can be sparsely selected both within and across groups. This technique uses all data relevant to a specific classification problem, but it does not leverage similarities among different classification problems.

In diagnosing brain diseases or disorders, Li et al. (2018) developed a deep domain adaptation neural network to improve the autism spectrum disorder classification by leveraging an autoencoder Vincent et al. (2010) trained on the data of a large number of healthy subjects. Ghafoorian et al. (2017) applied deep learning based domain adaptation to brain MRI for lesion segmentation with a convolutional neural network trained on a source domain of 280 patients and the last few layers fine-tuned on a target domain of 159 patients. Wachinger et al. (2016) proposed an instance re-weighting framework to improve the accuracy of Alzheimer's Disease (AD) diagnosis by making the source domain data to have similar distributions as

target domain data. Cheng et al. (2012, 2015a,b, 2017) proposed several workflows to perform domain adaptation to improve AD diagnosis accuracy by leveraging data of mild cognitive impairment, which is considered as the early stage of AD.

Separating general and specific features have also been applied in neuroimaging data analysis. For example, Pakravan & Shamsollahi (2018) used *Common Orthogonal Basis Extraction (COBE)* Zhou et al. (2015) to obtain subject-specific features by removing common components for the resting-state fMRI from HCP. On the extracted subject-specific features, behavioural prediction improved significantly compared to using the original features.

2.4 Summary

This chapter presented the fundamental theories and methods of domain-aware learning. The theoretical generalisation bounds of classical and multi-domain machine learning problems were reviewed, where the most significant difference is the domain distribution divergence in the multi-domain learning bounds. Then some popular methods were introduced by categories: feature mapping, domain-invariant classifier learning, domain adaptation neural networks, and feature separation, where the main focus is minimising the domain divergence as pointed out by the theory. Lastly, applications of multi-domain learning in neuroimaging data were briefly reviewed with the discussions on their limitations. The following chapters will present our research on domain-aware learning to improve the performance on neuroimaging data analysis.

Chapter 3

A Two-Stage Framework for Semi-Supervised Single-Source Adaptation

3.1 Introduction

In cognitive neuroscience, neuroimaging can help relate different cognitive functions to patterns of neural activity using functional magnetic resonance imaging (fMRI) (Ogawa et al., 1990). This often takes the form of a classification problem (Cox & Savoy, 2003), e.g. distinguishing between *brain conditions* associated with experimental stimuli. While fMRI produces volumes with the number of voxels in the order of 10^5, a typical experiment will have on the order of 10∼100 discrete trials, i.e. samples per cognitive condition/stimulus. This severely constrains the number of training examples available for the classifier. Moreover, neuroimaging data are noisy and contain a significant amount of physiological, respiratory, and mechanical artifacts, which requires robust modelling against noise (Aydore et al., 2019).

Domain adaptation is an attractive machine learning scheme and a potential solution to this problem. Chapter 2 has introduced existing domain adaptation works on neuroimaging data analysis. Despite the progress in the broad domain of neuroimaging, to the best of our knowledge, the existing state-of-the-art domain adaptation methods have not been systematically studied in whole-brain fMRI decoding for cognitive insights. This chapter proposes

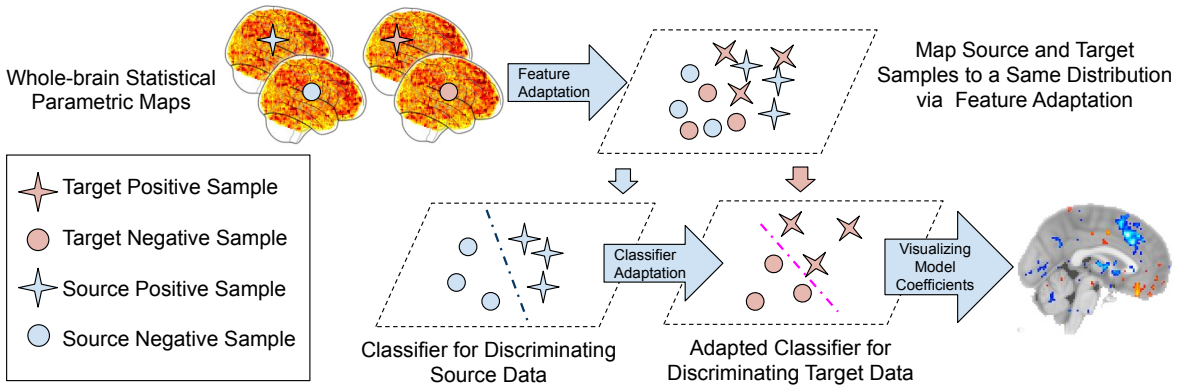


Figure 3.1: DawfMRI framework consists of two steps: feature adaptation, and classifier adaptation, e.g. via transfer component analysis, and cross-domain SVM, respectively. This is a semi-supervised domain adapation framework and therefore the target domain instances need to be partly labelled. Learnt model can be visualised on a brain atlas for interpretation.

a ***D****omain* ***a****daptation framework for* ***w****hole-brain* ***fMRI*** (DawfMRI) to improve the performance in a *target domain* classification problem with the help of *source domain* data. As shown in Fig. 3.1, this framework consist of two stages: feature adaptation and classifier adaptation, to reduce the distribution divergence and combined risk of the ideal-hypothesis in the generalisation bounds introduced in Theorem 2.14 and 2.16.

This framework enables systematic study of domain adaptation for whole-brain fMRI to evaluate and further develop feasible solutions. It can also help discover novel findings, understand how domain adaptation works in the context of neuroimaging, and identify key technical challenges. Our main contributions are twofold:

1. Methods: We formulated the DawfMRI framework consisting of two steps: 1) *feature adaptation*, and 2) *classifier adaptation*. Under this framework, we employed one state-of-the-art realisation of each step and evaluated all four possible variations.

2. Results: We designed experiments systematically using a collection of tasks from the OpenNeuro project. Results demonstrated a promising way of leveraging existing source data to improve brain decoding on target data. We discovered a plausible relationship between psychological similarity and adaptation effectiveness and revealed additional insights obtained via adaptation.

We also adopted a learning $\rightarrow$ evaluation $\rightarrow$ interpretation workflow, which will be followed

by the remaining empirical studies in Chapter 5 and 6 of this thesis. Our source code is available at: `https://github.com/sz144/DawfMRI`.

3.2 Methodology

3.2.1 A Feature-Classifier Adaptation Framework

We propose a two-stage domain adaptation framework for whole-brain fMRI (DawfMRI) as shown in Fig. 3.1. This framework consists of two steps: feature adaptation, and classifier adaptation.

- *Feature adaptation* is a domain adaptation scheme that utilises the source domain samples for target model training. The motivation is to leverage the samples from a related (source) domain when the information provided by the target domain samples is limited for training a good model. However, a classifier trained on source domain data will typically perform poorly on the target domain classification, which is assumed to be the result of domain feature distribution mismatch under the domain adaptation setting. The objective of feature adaptation is to minimize this mismatch by feature mapping or re-weighting (Pan & Yang, 2010). After performing feature adaptation, the adapted samples from source domain can be used as additional samples for training the target model.

- *Classifier adaptation* is another domain adaptation scheme that aims at improving the classifier performance in a target domain using the knowledge, such as coefficients or parameters, from a pre-trained classifier. The motivation is that when the information provided by target domain data is limited to train a good classifier, the discriminative information that a classifier learnt from source data can be leveraged to train a better target classifier.

There is a key difference between feature adaptation and classifier adaptation. The goal of feature adaptation is to make the source and target domain data similar. Classifier adaptation, in contrast, involves fitting a model to the source domain data, and using this model to

Table 3.1: Additional notations and descriptions used in this chapter.

Notation	Description
D	Input feature dimension
$\mathbf{D}_l^t$	Labelled target data
d	Output (lower) feature dimension
n	Number of samples
$\mathbf{U}$	Feature mapping/transformation matrix
$\mathbf{V}^s$	Source support vectors
$\mathbf{Z}^t$, $\mathbf{Z}^s$	Learned target/source representation

set priors for another model of the target domain. When feature adaptation is used without subsequent classifier adaptation, the feature-adapted source domain is used directly as if it contains additional training examples in the feature-adapted target domain. That is, a single model will be fit to the feature-adapted source and target domain data.

Each step of DawfMRI can be optional. If both of the two steps are skipped, we train a classifier directly on the whole-brain data, i.e. degenerating to standard machine learning setting. To study DawfMRI systematically, we employ a state-of-the-art method for each step in DawfMRI: transfer component analysis (TCA) (Pan et al., 2011) for feature adaptation, and cross-domain SVM (CDSVM) (Jiang et al., 2008) for classifier adaptation. Table 3.1 lists the key notations used for easy reference in the following presentation of these methods.

Feature Adaptation by TCA

TCA aims to find a feature mapping to minimize the mismatch between target and source distributions, and preserving the variance at the same time. For distribution mismatch, Maximum mean discrepancy (MMD) (Borgwardt et al., 2006) is used as the distribution mismatch metric. Given source domain data $\mathbf{X}^s \in \mathbb{R}^{D\times m_s}$, target domain data $\mathbf{X}^t \in \mathbb{R}^{D\times m_t}$, where m_s and m_t denote the number of samples of $\mathbf{X}^s$ and $\mathbf{X}^t$ respectively, and D denotes the input feature dimension, MMD between the two domains is

$$\begin{aligned}\mathrm{MMD}(\mathbf{X}^s,\mathbf{X}^t) &= \left\| \frac{1}{m_s}\sum_{i=1}^{m_s}(\mathbf{x}_i^s,\mathbf{X}^s) - \frac{1}{m_t}\sum_{i=1}^{m_t}(\mathbf{x}_i^t,\mathbf{X}^t) \right\|_{\mathcal{H}}^2 \\ &= \Big[\frac{1}{n_s^2}\sum_{i,j=1}^{m_s} k(\mathbf{x}_i^s,\mathbf{x}_j^s) - \frac{2}{m_s m_t}\sum_{i,j=1}^{m_s,m_t} k(\mathbf{x}_i^s,\mathbf{x}_j^t) + \frac{1}{n_t^2}\sum_{i,j=1}^{m_t} k(\mathbf{x}_i^t,\mathbf{x}_j^t)\Big],\end{aligned} \tag{3.1}$$

where $\mathcal{H}$ denotes a reproducing kernel Hilbert space (RKHS) (Berlinet & Thomas-Agnan, 2011), $k(\cdot,\cdot)$ denotes a kernel function, such as linear, Gaussian, and polynomial, $\mathbf{x}_i^s$ and $\mathbf{x}_j^t$ are the ith and jth sample of $\mathbf{X}^s$ and $\mathbf{X}^t$, respectively. TCA assumes that the domain difference is caused by marginal distribution mismatch, i.e, $p(\mathbf{X}^s) \neq p(\mathbf{X}^t)$. Hence, the objective is to learn new feature representations $\mathbf{Z}^s$ and $\mathbf{Z}^t$ by mapping the input data to a feature space where the MMD between the two domains is minimised, i.e., $p(\mathbf{Z}^s) \approx p(\mathbf{Z}^t)$. Equation (3.1) can be rewritten as $\text{MMD}(\mathbf{X}^s, \mathbf{X}^t) = \text{tr}(\mathbf{KL})$, where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is:

$$L_{ij} \begin{cases} \frac{1}{{m_s}^2} & \text{if} \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^s, \\ \frac{1}{{m_t}^2} & \text{if} \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^t, \\ -\frac{1}{m_s m_t} & \text{otherwise}, \end{cases} \tag{3.2}$$

and $n = m_s + m_t$. To minimize MMD, TCA employs a dimension reduction approach to learn a mapping matrix $\mathbf{U} \in \mathbb{R}^{n\times d}$ to map the samples from kernel space to a d-dimensional space ($d \ll m_s + m_t$) with a minimal difference in distribution. The MMD of the learnt space is

$$\text{tr}((\mathbf{K}\mathbf{U}\mathbf{U}^\top\mathbf{K})\mathbf{L}) = \text{tr}(\mathbf{U}^\top\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{U}). \tag{3.3}$$

The second objective is preserving the variance of the original input data. Similar to Kernel PCA (Schölkopf et al., 1998), covariance matrix $\text{cov}[\mathbf{U}^\top\mathbf{K}] = \mathbf{U}^\top\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{U}$ is used as the constraint of the TCA objective function, where $\mathbf{H} \in \mathbb{R}^{n\times n}$ is a centering matrix (Marden, 2014). Then the overall learning objective is

$$\begin{aligned} \min_{\mathbf{U}} \quad & \text{tr}(\mathbf{U}^\top\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{U}) + \lambda \cdot \text{tr}(\mathbf{U}^\top\mathbf{U}) \\ s.t. \quad & \mathbf{U}^\top\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{U} = \mathbf{I}_d, \quad \lambda > 0, \end{aligned} \tag{3.4}$$

where $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ is an identity matrix, λ is a tradeoff parameter for regularisation. Problem (3.4) can be optimised by the method of Lagrange multipliers and eigen-decomposition. The solution is the d smallest eigenvectors of $(\mathbf{K}\mathbf{L}\mathbf{K} + \lambda\mathbf{I})(\mathbf{K}\mathbf{H}\mathbf{K})^{-1}$.

Classifier Adaptation by CDSVM

CDSVM is an SVM classifier that utilises the source support vectors learnt by a standard SVM on source domain samples to find a better decision boundary for target samples. It re-weights each source support vector according to its average distance to the target (training) feature vectors, and then the target classifier will be trained with the target training samples and re-weighted source support vectors. We learn the decision function of CDSVM $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ by optimizing the following objective:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi_i + C\sum_{j=1}^{K}\sigma(\mathbf{v}_j^s, \mathbf{D}_l^t)\bar{\xi}_j \\ s.t. \quad & y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall(\mathbf{x}_i, y_i) \in \mathbf{D}_l^t, \\ & y_j^s f(\mathbf{v}_j^s) \geq 1 - \bar{\xi}_j, \quad \bar{\xi}_j \geq 0, \quad \forall(\mathbf{v}_j^s, y_j^s) \in \mathbf{V}^s, \\ & \sigma(\mathbf{v}_j^s, \mathbf{D}_l^t) = \frac{1}{M}\sum_{(\mathbf{x}_i, y_i)\in\mathbf{D}_l^t} \exp(-\beta\|\mathbf{v}_j^s - \mathbf{x}_i\|_2^2). \end{aligned} \tag{3.5}$$

$\mathbf{D}_l^t$ represents the labelled (training) target domain data. $\mathbf{V}^s$ denotes a matrix composed of all source support vectors. M is the number of samples in $\mathbf{D}_l^t$. K is the number of source support vectors in $\mathbf{V}^s$. $\mathbf{v}_j^s$ is the jth source support vector in $\mathbf{V}^s$. ξ_i and $\bar{\xi}_j$ are slack variables for the ith target feature vector and jth source support vector respectively. C is a hyperparameter controlling the trade-off between the slack variable penalty and the SVM soft margin. $\sigma(\mathbf{v}_j^s, \mathbf{D}_l^t)$ is a function that evaluates the distance between $\mathbf{v}_j^s$ and $\mathbf{D}_l^t$. β is a hyperparameter controlling the influence of the source support vectors. Larger value of β leads to less influence of source support vectors, and vice versa.

3.2.2 Method for Model Coefficients Interpretation

Apart from model training and evaluation, interpretation is another important part of neuroimaging data analysis. The final classifier coefficients (or weights) indicate the significance of the corresponding features for a classification problem. Visualising them in the brain voxel space can help us gain some insights into which areas contribute more to prediction performance. To achieve this, we chose a linear kernel in TCA, SVM and CDSVM and developed

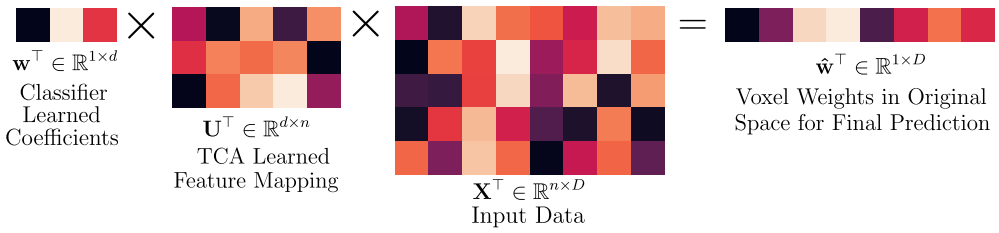


Figure 3.2: Mapping classifier coefficients $\mathbf{w} \in \mathbb{R}^{d\times 1}$ back to voxel weights $\hat{\mathbf{w}} \in \mathbb{R}^{D\times 1}$ in the voxel feature space.

a method to map the classifier coefficients back to voxel weights in the original brain voxel space for interpretation, as shown in Fig. 3.2.

In the following, we detail how to map the model coefficients of TCA+SVM (or TCA+ CDSVM) to the original voxel space.

Using the same notations above, we have $\mathbf{X} \in \mathbb{R}^{D\times n}$ (or $\mathbf{X} \in \mathbb{R}^{(D+1)\times n}$ if the bias/intercept is fitted) composed of target and source data, TCA learnt a feature transformation $\mathbf{U} \in \mathbb{R}^{n\times d}$, and SVM has learnt coefficients $\mathbf{w} \in \mathbb{R}^d$. According to Eq. (3.3), the TCA features can be represented as $\mathbf{Z} = \mathbf{U}^\top\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{d\times n}$. For a sample $\mathbf{x} \in \mathbb{R}^D$ (or $\mathbf{x} \in \mathbb{R}^{(D+1)}$ if the bias/intercept is fitted), its transformed feature $\mathbf{z} = \mathbf{U}^\top\mathbf{X}^\top\mathbf{x} \in \mathbb{R}^{d\times 1}$. The predicted class is

$$\hat{y} = sgn(\mathbf{w}^\top\mathbf{z}) = sgn(\mathbf{w}^\top\mathbf{U}^\top\mathbf{X}^\top\mathbf{x}), \tag{3.6}$$

where sgn is the sign function (1 for positive values, -1 for negative values). Let $\hat{\mathbf{w}}^\top = \mathbf{w}^\top\mathbf{U}^\top\mathbf{X}^\top \in \mathbb{R}^{1\times D}$ (or $\in \mathbb{R}^{1\times(D+1)}$ if the bias/intercept is fitted), we obtain

$$\hat{y} = sgn(\hat{\mathbf{w}}^\top\mathbf{x}). \tag{3.7}$$

Hence, $\hat{\mathbf{w}}$ contains the weights corresponding to each voxel in the whole brain space for final prediction. When a linear kernel svm, e.g. CDSVM, is used only, the prediction for an instance $\mathbf{x}$ can be obtained via

$$\hat{y} = sgn(\mathbf{w}^\top\mathbf{z}) = sgn(\mathbf{w}^\top\mathbf{X}^\top\mathbf{x}), \tag{3.8}$$

where $\mathbf{w} \in \mathbb{R}^n$. Similarly, Eq. (3.7) can be obtained by letting $\hat{\mathbf{w}}^\top = \mathbf{w}^\top\mathbf{X}^\top \in \mathbb{R}^{1\times D}$ (or

$\in \mathbb{R}^{1\times(D+1)}$ if the bias/intercept is fitted), and $\hat{\mathbf{w}}$ contains the weights corresponding to the original feature for final prediction.

3.3 Materials and Experiments

3.3.1 OpenNeuro Data and Pre-processing

Dataset

Eleven datasets[1] from OpenNeuro were used in the experiments. Table 3.2 lists the details of the selected datasets. Each dataset is from a cognitive experiment. Each sample was processed with the protocol in (Poldrack et al., 2013) to obtain the Z-score statistical parametric map (SPM) (Friston et al., 1994, 1998; Gorgolewski et al., 2015) of standard MNI152 template size $91 \times 109 \times 91$, which is then reduced to a vector of size $211,106$ containing only the voxels within the brains. The contrasts associated with each task represent differences between the primary tasks and baseline conditions. We conducted our experiments using the single contrast per task. The contrasts used are also reported in Table 3.2.

Pre-processing Pipeline

To process the data from OpenNeuro, we implemented a standard preprocessing pipeline using FSL (Jenkinson et al., 2012) based on the processing stream[2] implementation (Poldrack et al., 2013). The output of one step will be the input of next step. As shown in Table 3.3, the pipeline has five steps:

1. Perform motion correction on the BOLD signal sequences from OpenNeuro using MCFLIRT (FSL).

2. Perform brain extraction using BET (FSL).

3. Perform first-level analysis to generate Z-score statistical parametric maps (SPMs) (Friston et al., 1994, 1998) of contrasts for each experiment condition using FEAT (FSL). FSL design files are generated from the onsets files using the custom code.

[1] We use the data in legacy format available at `https://legacy.openfmri.org`

[2] `https://github.com/poldrack/openfmri`

Table 3.2: List of OpenNeuro datasets used in our experiments (ACN denotes accession number. #Sample denotes the number of samples for each dataset. Abbr denotes abbreviations, which are used in the rest of this chapter for easy reference).

ACN	#Sample	Task & Contrast Description	Abbr
ds001	32	Balloon analog risk task: Parametric pump effect vs. control	BART
ds002	34	Probabilistic classification: Task vs. baseline	CT1
ds002	34	Deterministic classification: Feedback vs. baseline	CT2
ds002	34	Mixed event-related probe: Task vs. baseline	CT3
ds003	13	Rhyme judgement: Task vs. baseline	RJT
ds005	32	Mixed-gambles task: Parametric gain response	MGT
ds007	40	Stop signal task: Letter classification vs. baseline	SST1
ds007	38	Stop signal task: Letter naming vs. baseline	SST2
ds007	39	Stop signal task: Pseudoword naming vs. baseline	SST3
ds101	42	Simon task: Incorrect vs. correct	ST
ds102	52	Flanker task: Incongruent vs. congruent	FT

Table 3.3: Preprocessing pipeline for the selected OpenNeuro data.

Steps	Operation Description	Tools Used
1	Motion correction	MCFLIRT (FSL)
2	Brain extraction	BET (FSL)
3	Within-run statistical analysis	FEAT (FSL)
4	Alignment of Z statistic maps	featregapply (FSL)
5	Masking (MNI152 T1 2mm brain mask) and Vectorisation	Nibabel

4. Align the spatially normalised SPMs obtained in Step 3 with the MNI152 standard atlas of size $91 \times 109 \times 91$ using featregapply (FSL).

5. Vectorise the voxels from the SPMs that fall within the standard MNI152 brain mask (distributed with FSL) using the Python package Nibabel (Brett et al., 2017).

The contrasts associated with each task represent differences between the primary tasks and some baseline conditions. Rather than considering the influence of various baseline conditions, we conducted our experiments using the same single contrast per task as (Poldrack et al., 2013). The contrasts used are also reported in Table 3.2.

3.3.2 Experiment Settings and Evaluation Methods

Our experiments will focus on using domain adaptation to improve performance on challenging binary classification problems that require distinguishing brain states associated with

different cognitive tasks. Thus, we only considered the most basic scenario, specifically, both target and source classification problems are binary. We studied DawfMRI in the setting of one-to-one domain adaptation only. This means that one target domain will be supplemented by only one set of source domain data.

Algorithm Setup

Four possible variations of the DawfMRI framework, including one non-adaptation algorithm: 1) SVM, and three adaptation algorithms: 2) CDSVM,3) TCA+SVM, 4) TCA+CDSVM, are evaluated.

For TCA+SVM, a SVM was fit to the both source and target training samples after performing TCA. For CDSVM and TCA+CDSVM, a SVM was fit to the source samples, and then the source support vectors were used as additional input for training CDSVM on the target samples.

As mentioned in Sec. 3.2.2, we chose a linear kernel in TCA, SVM, and CDSVM for easy interpretation. We optimised hyperparameters on regular grids of log scale for each algorithm, with a step size of one in the exponent. We searched for the best C and μ values within the range $[10^{-3}, 10^{3}]$ and $[10^{-5}, 10^{5}]$ for SVM and TCA, respectively. For CDSVM, we grid-searched for the best combination of C and β values in $[10^{-3}, 10^{3}]$. We also varied the feature dimension of TCA output from 2 to 100 (2, 10, 20, 50 and 100), and optimize the relevant algorithms with the best feature dimensions.

Target and Source Domain Setup

Each task (listed in Table 3.2) is associated with data collected from a number of participants over one or two scanner runs. An SPM expressing a particular contrast between task and baseline exists for each run for each subject. These SPMs comprise the set of possible training examples, and the tasks serve as the category labels. Each pair of tasks is a domain, and our problem is binary classification between two tasks. Each domain can be used as a target (the primary problem that we want to improve performance on) or a source (the secondary problem from which we want to leverage knowledge to help better solve the primary problem). We anticipate the classification problems with lower prediction accuracy to have

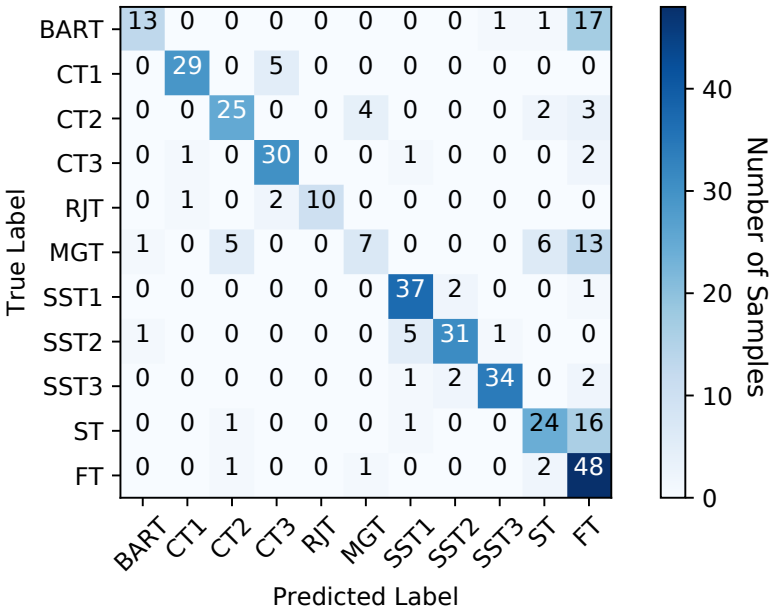


Figure 3.3: Multi-class classification confusion matrix for linear SVM performance on the whole-brain SPMs. Entry (i, j) in the confusion matrix is the number of observations actually in task i, but predicted to be in task j. Four most challenging pairs of classification tasks (BART vs FT, MGT vs CT2, MGT vs ST, and MGT vs FT) were selected as target domains to perform domain adaptation.

higher potential of improvement via domain adaptation. Therefore, we selected four most challenging domains, with highly confusable pairs of tasks:

1. BART (32 samples) vs FT (52 samples),
2. MGT (32 samples) vs CT2 (34 samples),
3. MGT (32 samples) vs ST (42 samples),
4. MGT (32 samples) vs FT (52 samples)

These domains were identified by performing multi-class classification. Specifically, we used a linear SVM to classify all eleven classes of the whole-brain SPMs. Figure 3.3 shows the 10-fold cross-validation results as a confusion matrix, where an entry (i, j) is the number of observations actually in task i, but predicted to be in task j. This allows us to identify the most confusable pairs. The results indicate that the *balloon analog risk task* (BART) and *mixed-gambles task* (MGT) were often confused with other tasks. BART was misclassified as *flanker task* (FT) more than half the time. MGT was the least accurately classified overall,

of which the samples were often misclassified as the *deterministic classification* (CT2), *Simon task* (ST), or *flanker task* (FT). We then identified the four aforementioned pairs of tasks as target domains to focus on. These selected pairs are confirmed later to be those benefiting the most from domain adaptation in our adaptation effectiveness study (Sec. 3.3.3).

Source selection and class labels: When using one pair of tasks as the target domain, the remaining nine tasks are combined pairwise to give 36 unique pairs, each of which is a candidate source domain. For each pair, one task is labelled 1 and the remaining task is labelled -1, also called positive and negative classes, respectively. There are two ways to match source domain labels with target domains labels, i.e., 1 with 1 and -1 with -1, or 1 with -1 and -1 with 1. We studied both cases.

Evaluation Methods

We performed 10×10-fold cross-validation (CV) evaluation. All training samples were sampled uniformly at random for cross validation. CV was only applied to target domain samples. Source domain samples were all used for training when performing domain adaptation. We will report the mean classification accuracy with standard derivations for performance evaluation. To study the statistical significance of the results obtained by adaptation algorithms compared to those by non-adaptation algorithms, we will report the p-values of paired t-tests.

3.3.3 Classification Results

Results on Different Target Domains

Figure 3.4 shows the 10-CV classification results across the four different target domains. For adaptation algorithms, we tested all possible source domains to report the best results in Fig. 3.4, with the corresponding sources indicated in the bars. The best results on the four target domain were obtained by TCA+SVM and TCA+CDSVM. TVA+SVM and TCA+CDSVM outperformed the non-adaptation algorithms consistently and significantly (maximum p-value < 0.0001 in paired t-test). The largest accuracy improvement was obtained by TCA+CDSVM on BART vs FT with the source SST2 vs CT2. This improves over the SVM by **10.47%** (**from 77.26% to 87.73%**).

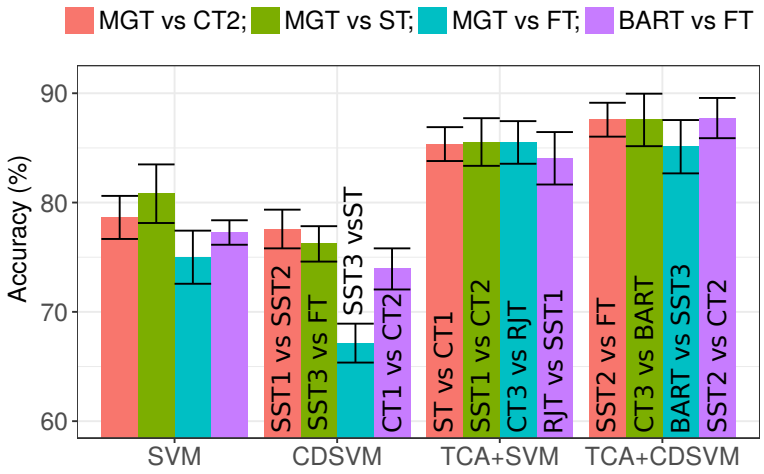


Figure 3.4: Classification accuracy of four DawfMRI variations (x-axis) on the four target domains (coloured bars). Error bars indicate the standard derivations. Adaptation algorithms use the best source domains, as indicated in the bars.

Effectiveness of Each DawfMRI Step

In feature adaptation, TCA can extract features with lower dimension from whole-brain SPMs. Moreover, features extracted by TCA are common and useful across source and target domains. As shown in Figures 3.4, TCA+SVM outperformed non-adaptation algorithms with appropriate sources. This indicates that by performing TCA, samples from appropriate source domains can be used as additional training data for target domain.

In classifier adaptation, CDSVM can improve the accuracy when combined with TCA. However, by comparing the results in Figs. 3.4, SVM outperformed CDSVM consistently, while TCA+CDSVM did not outperform TCA+SVM consistently. This indicates that the effectiveness of CDSVM tends to be related to the distribution distances between domains.

Sensitivity to Source Domain

Figure 3.5 summarises the *adaptation effectiveness* of different source domains over the four different target domains, which are sorted by the largest accuracy improvements of TCA+CDSVM over SVM. The accuracy were improved on 123 out of 144 possible target and source combinations. However, the improvement varies across different source domains. It shows that the effectiveness of domain adaptation was significantly affected by the source domain, which motivated our target domain selection strategy in Sec. 3.3.2. We also observed

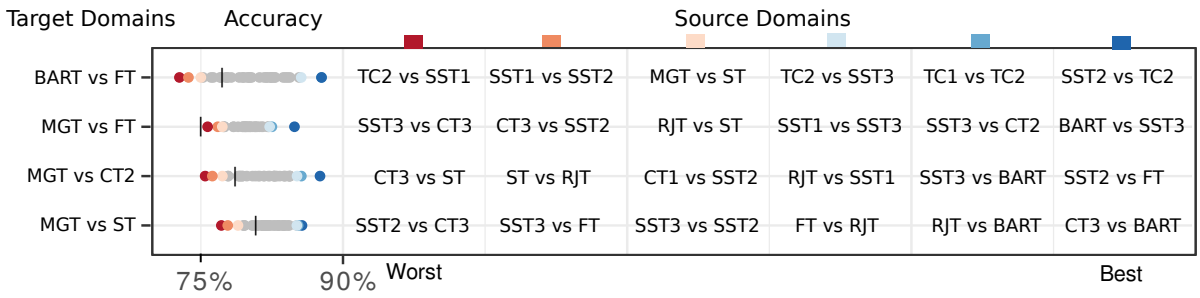


Figure 3.5: Adaptation effectiveness of TCA+CDSVM (coloured dots) over SVM (black vertical bars) across all target domains, with 10-fold cross-validation. Target domains are sorted with respect to the maximum improvement. The top three and bottom three source domains are listed on the right half, in the order of the worst, the second worst, the third worst, the third best, the second best, and the best, from left to right.

that SVM outperformed TCA+CDSVM in some cases, which is called "negative transfer" (Pan & Yang, 2010).

3.4 Discussion

This section will further analyse DawfMRI with two objectives to facilitate further discussion: 1) exploring whether adaptation effectiveness is related to psychological similarities between tasks, 2) understanding how domain adaptation improves brain decoding by visualising the model coefficients.

3.4.1 Psychological Interpretation of Source Domain Effectiveness

Domain adaptation effectiveness is closely related to meaningful relationships between the target and source domains. On the other hand, psychological experiments are intrinsically related by the cognitive mechanisms that support the ability to perform the tasks. Hence, we expect the cognitive similarity between a set of tasks to be predictive of whether or not domain adaptation will be effective.

Table 3.4: Statistics of the four psychological similarity features used for logistic model training, which are target domain similarity (TDSim), source domain similarity (SDSim), cross-domain similarity (CDSim) and target domain SVM accuracy (TDSVM_Acc).

	CDSim	TDSim	SDSim	TDSVM_Acc
Min	-1.78	-1.02	-1.21	-1.60
Median	-0.05	-0.59	-0.36	-0.026
Mean	0.00	0.00	0.00	0.00
Max	3.47	1.93	2.73	1.33

Table 3.5: Logistic model learning for studying the relationship between psychological similarity and adaptation effectiveness. The model was regressed on four variables, TDSim, SDSim, CDSim, and TDSVM_Acc, for predicting improved or not improved.

	Beta	Standard Error	t-value	p-value
(intercept)	-0.085	0.097	-0.88	0.38
TDSim	-0.10	0.11	-0.98	0.32
SDSim	-0.20	0.11	-1.89	0.057
CDSim	0.26	0.11	2.38	0.017
TDSVM_Acc	-0.80	0.10	-7.75	9.15e-15

Psychological Similarity Study

We explored a potential relationship between psychological similarity and adaptation effectiveness by modelling the probability of domain adaptation with TCA+CDSVM improving prediction accuracy as a function of the psychological similarity between tasks within and across domains.

To estimate the psychological similarity, we associated each task with a set of cognitive functions that it relies on. The associations were defined by referring to the cognitive concepts in Cognitive Atlas[3] (Poldrack et al., 2011). Out of the 11 tasks in Table 3.2, 10 were associated with a number of cognitive functions (min = 5, max = 13, median = 7). The *mixed event related probe* (CT3) had an incomplete entry so it was excluded from these and further studies. There were 42 functions in total, and each task was represented as a binary feature vector. The psychological similarity between each pair of tasks was computed as the cosine similarity between their feature vectors.

Because each domain is composed of a pair of tasks, the similarity between the target and

[3] `http://www.cognitiveatlas.org/`

source domains in each model is associated with four pairwise task similarities. The overall psychological similarity between each pair of domains was estimated by averaging these four pairwise similarities. We will refer to this estimate as the *Cross-Domain Similarity* (CDSim). Moreover, the similarity between the two tasks of the target domain is denoted as the *Target-Domain Similarity* (TDSim), and the similarity between the two tasks of the source domain is denoted as the *Source-Domain Similarity* (SDSim).

This study focused on 504 target and source combinations of the classification problems with no more than 90% in accuracy obtained by whole-brain SVM. Of these 504 models, 261 were improved. We labelled them as 'improved' or 'not improved' to train a logistic model. This binary outcome was regressed on four variables, CDSim, TDSim, SDSim, and TDSVM_Acc (the accuracy of the standard SVM in the target domain). The variables were standardised to have a mean of zero and standard deviation of one before fitting the model. Table 3.4 lists the statistics of the four variables.

Table 3.5 reports the learning outcome of the logistic model, where increasing CDSim increased the probability of improved accuracy. This relationship between psychological similarity and adaptation effectiveness is important. On the one hand, it is consistent with how these adaptation methods are meant to work. On the other hand, it suggests that it may be possible to predict the adaptation effectiveness in advance, without resorting to a post-hoc selection of the source domain through trial and error.

Source Selection Validation

We also adopted a leave-one-target-domain-out strategy to learn to "select" an appropriate source domain. For the hold-out target domain, we selected the source domain with the highest likelihood of accuracy improvement given by the logistic model. Then we compared the real improvement of using the selected source data against random source selection, i.e., the mean improvement, for TCA+SVM and TCA+CDSVM. Results showed that psychological similarity based source selection led to 0.0068, 0.0065, and 0.0371 higher classification accuracy than random selection. Therefore, it can help source selection.

In addition, we did the same analysis to TCA+SVM to compare MMD-based source selection with random source selection. We computed the MMDs for all possible target

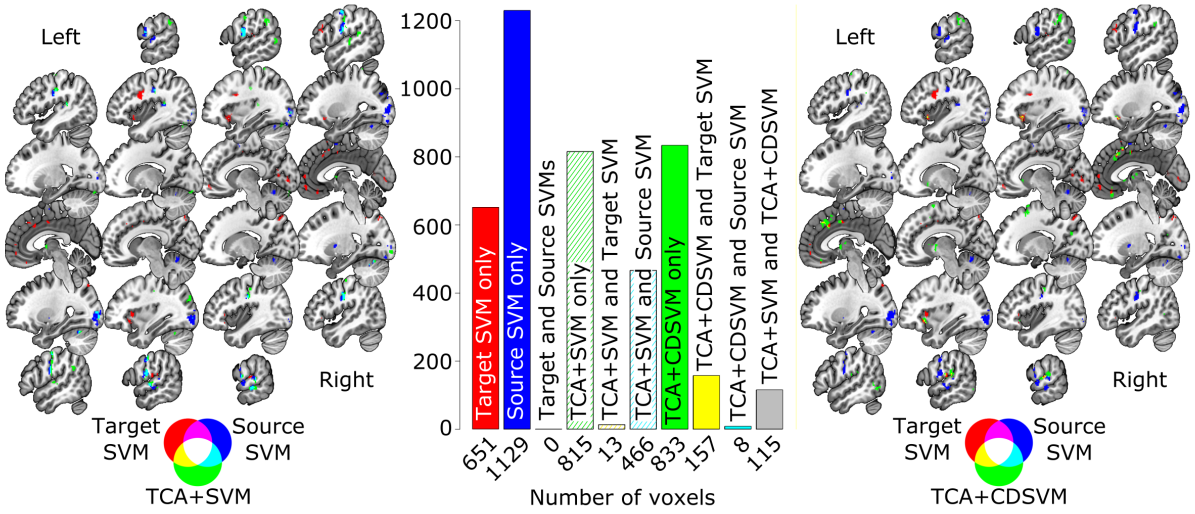


Figure 3.6: Visualisation of the voxels with top 1% weight magnitude and occurring in clusters of at least 20 voxels in the four models: target SVM, source SVM, TCA+SVM, and TCA+CDSVM. Numbers of distinct and overlapped voxels identified by the models are shown in the middle bars.

and source combinations using Eq. (3.3) after performing TCA, as well as the accuracy of respective TCA+SVM. Results showed that on average, selecting source domains with the smallest MMD can achieve 0.064 higher accuracy than random source selection.

3.4.2 Neural Decoding Visualisation and Cognitive Interpretation

It is also important to understand why a model performs well and which brain networks are particularly important, e.g. for advances in cognitive neuroscience and understanding neural disorder physiology. Exploring what information tends to emerge through domain adaptation will provide insights into how these methods work and what cognitive similarity is being leveraged through domain adaptation. Therefore, we carried out two studies to examine where the important voxels are in the brain, and how much the voxel sets in the target domain, source domain, and adapted models overlap.

Model Overlapping Study I

We firstly studied the case of a balanced target classification problem, which is the *mixed-gambles task* (MGT) vs *deterministic classification task* (CT2). *Pseudoword naming stop*

signal task (SST3) vs *flanker task* (FT) is used as source, which showed the biggest improvement in classification accuracy for the selected target problem. Figure 3.6 shows the voxels with the top 1% weight magnitude in the four models (target SVM, source SVM, TCA+SVM, and TCA+CDSVM) and occurring in clusters of at least 20 voxels. The target and source domain SVMs place their important voxels in completely different areas. Not only is there no overlap, but the supra-threshold voxels in each model are sampled from different lobes of the brain: the target domain SVM is associated primarily with supra-threshold voxels in the frontal lobes, while the source domain SVM is associated primarily with supra-threshold voxels in the occipital lobe and sensory motor cortex. The distribution of coefficients is so different between source and target models, but adaptation can be nevertheless very effective.

We then overlaid the adapted models from TCA+SVM and TCA+CDSVM. TCA+SVM has substantial overlap with the source domain SVM, and nearly no overlap with the target domain SVM. This substantial overlap, however, is only about 1/3 of the supra-threshold voxels in the TCA+SVM model. The remaining 2/3 are completely distinct from either the target or source models, indicating that information from the source domain has revealed a different dimension along to which to dissociate the tasks in the target domain than was apparent in the target data in isolation.

This general pattern is echoed in the model adapted with TCA+CDSVM, except that in this case there is virtually no overlap with the source domain and there is instead modest overlap with the target domain. Again, the adapted model is largely associated with supra-threshold voxels that do not overlap with either the target or source SVMs. Thus, the adaptation procedure has provided additional insights into the classification problem, showing the exploited information to be more than the simple sum of information from the target and source domain models.

Model Overlapping Study II

We further analysed the overlapped important voxels (with top 1% weight magnitude) in the four aforementioned models for 142 different target-source pairs where TDSVM_Acc$\leq$ 90% and TCA+CDSVM leading to at least 3% improvement. We examined the number of (overlapped) voxels for all 15 possible combinations of the four models, including individual

Figure 3.7: The overlapped important voxels for all 15 possible combinations (y-axis) of the four models: target SVM, source SVM, TCA+SVM, and TCA+CDSVM. The x-axis denotes 142 different target-source pairs where TDSVM_Acc$\leq$ 90% and TCA+CDSVM leading to at least 3% improvement. The (overlapped) voxel numbers for the 15 model combinations form a 15-element vector for each target-source pair. This vector is normalised to unit length and visualised as a column, where the normalised number of voxels (from 0 to 1) is denoted as the colour (from blue to yellow) in the heatmap.

models. We formed a 15-element vector for each target-source pair with 15 such numbers and then normalised it to unit length. Figure 3.7 depicts the normalised vectors as columns for all 142 target-source pairs, labelled with the corresponding model(s).

There is a clear effect that each model identifies a fairly distinct set of voxels (rows 1, 2, 4, and 8), which are more than the overlapped voxels between non-adaptation and adaptation models (rows 5, 6, 9 and 10). On the other hand, by comparing the results shown in rows of 5, 6, 9 and 10, TCA+SVM overlaps with the source domain SVM much more than with the target domain SVM, and the opposite is true for TCA+CDSVM. This again confirmed that adaptation is exploiting information additional to the target and source domain models, and different adaptation schemes are exploiting different information.

3.4.3 Technical Challenges

Based on the experimental results, we can see two technical challenges in DawfMRI. One is *how to select a good source domain automatically (without exhaustive testing) to reduce or*

even avoid "negative transfer", as observed in our experiments. The plausible relationship between psychological similarity and adaptation effectiveness in Sec 3.4.1 can lead to a better than random solution. However, it is not the optimal selection of sources. The other challenge is *how to make use of multiple source domains to further improve the classification performance*. We need to carefully leverage the positive effects from each source domain while minimising the potential negative impacts. This needs a smart, adaptive procedure to be introduced. We consider both challenges are important directions to explore in the future.

3.5 Summary

In this chapter, a two-stage domain adaptation framework is proposed for whole-brain fMRI (DawfMRI), which consists of two key steps: feature adaptation and classifier adaptation, to reduce the distribution divergence and combined. Two state-of-the-art algorithms TCA and CDSVM are employed for each step of DawfMRI, respectively. Experimental studies of two non-adaptation algorithms and six adaptation algorithms on task-based whole-brain fMRI from eleven OpenNeuro datasets shown that DawfMRI can significantly improve the learning performance for challenging binary classification problems. Additionally "negative transfer" was observed in the experiments, indicating that domain adaptation does not always give better performance and should be used with care. Furthermore, we discovered a plausible relationship between psychological similarity and adaptation effectiveness, and interpreted how the models provide additional insights. Finally, we pointed out two important research directions to pursue in future work.

Chapter 4

Dependence-Based Multi-Domain Generalisation Theory

Chapter 3 pointed out an important research question: multi-source domain adaptation. For this problem, two $\mathcal{H}\Delta\mathcal{H}$-based generalisation bounds on VC dimension have been introduced in Chapter 2, where the tighter bound (Theorem 2.18) is proved by viewing the mixture of source domains as a single source domain. This strategy is usually called "source combine". However, it has been shown in many practical studies that the source combine strategy can hardly outperform aligning each source-target pair (Theorem 2.17). This chapter will present further theoretical analysis for multi-source domain adaptation generalisation bounds and then derive a statistical dependence based bound, which will be the theoretical support for the following two chapters.

4.1 "One vs One" and "One vs Rest" Bounds

In multi-source domain adapattion, there has been many algorithms proposed by following Theorem 2.17 and 2.18, which have been introduced in Chapter 2. There is another multi-source domain adaption network introduced in Chapter 2: *Moment Matching for Multi-Source Domain Adaptation* (M^3SDA) (Peng et al., 2019), which optimises the sum of divergence between an arbitrary pair of domains, i.e. the divergence between two source domains are also considered, instead of optimising the sum of divergences between each pair

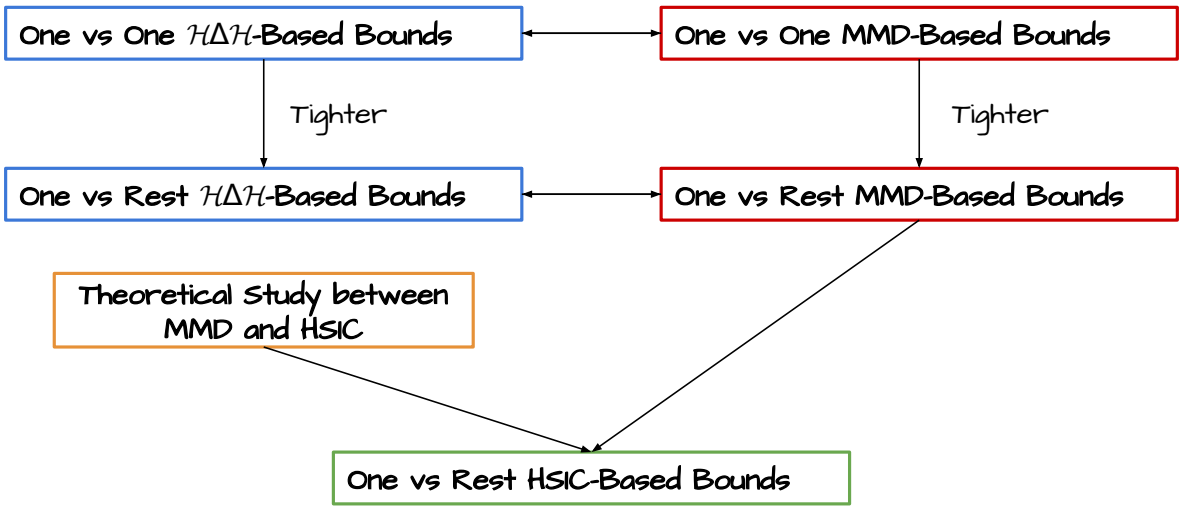


Figure 4.1: Relationships of the generalisation bounds derived in this chapter, where the statistical dependence/HSIC-based bound is the proposed and recommended one.

of source target domain, because "it is difficult to align a target domain with every source domain without aligning source domains together". Using the example of three domains A, B and C in Chapter 2 again, the divergence mesurment in the objective function of M^{3}SDA is $|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|+|\mathrm{B}-\mathrm{C}|$. However, the optimal $|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|+|\mathrm{B}-\mathrm{C}|$ cannot guarantee $|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|$ (divergence in Theorem 2.17) to be optimal. Therefore, unlike the authors claimed, M^{3}SDA does not optimise the "one vs target" generalisation bound of Theorem 2.17. Inspired by the multi-class classification problems, the objective of M^{3}SDA can be summarised as an "one vs one" divergence, and the following theorem can be derived:

Theorem 4.1 (One-vs-one $\mathcal{H}\Delta\mathcal{H}$ multi-source bound)**.** *Let $\mathcal{H}$ be a hypothesis space of VC dimension* $\mathrm{VC}(\mathcal{H})$*. Let $\mathbf{X}_j$ be labelled samples of size $\beta_j m$ drawn from distribution $\mathcal{D}_j$ with weight α_j for $j \in \{1,\ldots,J+1\}$, where $\alpha_{J+1}=1/J$, and $\beta_{J+1}=1$, let $\mathbf{X}^t=\mathbf{X}_{J+1}$, $\mathbf{X}^s=[\mathbf{X}_1,\ldots,\mathbf{X}_J]$, then for any $\delta\in(0,1)$, with the probability at least $1-\delta$, for every $h\in\mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \hat{\mathrm{R}}_s(h) + \sum_{i,j=1,i\neq j}^{J+1} (\alpha_i+\alpha_j)\Big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}_i,\mathbf{X}_j)+\lambda_{i,j}\Big) + \mathrm{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right) \tag{4.1}$$

where $\lambda_{i,j}=\min_{h\in\mathcal{H}}\{\mathrm{R}_i(h)+\mathrm{R}_j(h)\}$, and $\mathrm{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right)$ is the complexity term.

Proof. Let $h^*_{i,j} = \arg\min_{h\in\mathcal{H}}\{\mathrm{R}_i(h) + \mathrm{R}_j(h)\}$. Then

$$
\begin{aligned}
&|\mathrm{R}_s(h) - \mathrm{R}_t(h)| \\
&\quad = \Big|\sum_{j=1}^{J} \alpha_j \mathrm{R}_j(h) - \mathrm{R}_t(h)\Big| \leq \sum_{j=1}^{J} \alpha_j \big|\mathrm{R}_j(h) - \mathrm{R}_t(h)\big| \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \sum_{i=1,i\neq j}^{J} \big|\mathrm{R}_i(h) - \mathrm{R}_j(h)\big| \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \sum_{i=1,i\neq j}^{J} \big(\big|\mathrm{R}_i(h) - \mathrm{R}_i(h, h^*_{i,j})\big| + \big|\mathrm{R}_i(h, h^*_{i,j}) - \mathrm{R}_j(h, h^*_{i,j})\big| + \big|\mathrm{R}_j(h, h^*_{i,j}) - \mathrm{R}_j(h)\big|\big) \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \sum_{i=1,i\neq j}^{J} \big(\mathrm{R}_i(h^*_{i,j}) + \big|\mathrm{R}_i(h, h^*_{i,j}) - \mathrm{R}_j(h, h^*_{i,j})\big| + \mathrm{R}_j(h^*_{i,j})\big) \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \sum_{i=1,i\neq j}^{J} \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_i, \mathcal{D}_j) + \lambda_{i,j}\big).
\end{aligned}
\tag{4.2}
$$

Now with Eq. (4.2) and Hoeffdings inequality, for any $\delta \in (0, 1)$, with the probability $1 - \delta$,

$$
\begin{aligned}
\mathrm{R}_t(h) \leq& \mathrm{R}_s(h) + \sum_{j=1}^{J+1} \alpha_j \sum_{i=1,i\neq j}^{J} \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_i, \mathcal{D}_j) + \lambda_{i,j}\big) \\
\leq& \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1} \alpha_j \sum_{i=1,i\neq j}^{J} \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}_i, \mathbf{X}_j) + \lambda_{i,j}\big) + \mathrm{O}\Big(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\Big)
\end{aligned}
\tag{4.3}
$$

where $\lambda_{i,j} = \min_{h\in\mathcal{H}}\{\mathrm{R}_i(h) + \mathrm{R}_j(h)\}$. This completes the proof. □

Theorem 4.1 can also be proved by using Theorem 2.17 directly:

$$
\begin{aligned}
\mathrm{R}_t(h) \leq& \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J} \alpha_j \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}_j, \mathbf{X}^t) + \lambda_j\big) + \mathrm{O}\Big(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\Big) \\
\leq& \hat{\mathrm{R}}_s(h) + \sum_{i,j=1,i\neq j}^{J+1} \alpha_j \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}_i, \mathbf{X}_j) + \lambda_{i,j}\big) + \mathrm{O}\Big(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\Big)
\end{aligned}
\tag{4.4}
$$

Theorem 4.1 explains the exact objectives of M^{3}SDA as a generalisation bound. The only difference is the $\mathcal{H}\Delta\mathcal{H}$-divergence is replaced by moment distance in (Peng et al., 2019).

By borrowing the concept of one-vs-one from multi-class classification, this bound can be called as one-vs-one multi-source generalisation bound. By leveraging the theoretical results in Theorem 2.16, a MMD-based one-vs-one bound for multi-source domain adaptation can be proved as the following theorem.

Theorem 4.2 (One-vs-one MMD multi-source bound)**.** *Let $\mathcal{H}$ be a hypothesis space. Let $\mathbf{X}_j$ be labelled samples of size $\beta_j m$ drawn from distribution $\mathcal{D}_j$ with weight α_j for $j \in \{1, \ldots, J+1\}$, where $\alpha_{J+1} = 1/J$, and $\beta_{J+1} = 1$, let $\mathbf{X}^t = \mathbf{X}_{J+1}$, $\mathbf{X}^s = [\mathbf{X}_1, \ldots, \mathbf{X}_J]$, and $\mathbf{K}_j$ be the kernel matrix of $\mathbf{X}_j$ with kernel function $k(\cdot, \cdot)$ then for any $\delta \in (0, 1)$, with the probability at least $1 - \delta$, for every $h \in \mathcal{H}$:*

$$\begin{aligned} \mathrm{R}_t(h) \leq & \hat{\mathrm{R}}_s(h) + \sum_{i,j=1, i\neq j}^{J+1} (\alpha_i + \alpha_j)\Big(\hat{d}_{\mathrm{MMD}}(\mathbf{X}_i, \mathbf{X}_j) + \frac{2}{\beta_i m}\mathbb{E}_{\mathbf{X}_i}\big[\sqrt{\mathrm{tr}(\mathbf{K}_i)}\big] + \\ & \frac{2}{\beta_j m}\mathbb{E}_{\mathbf{X}_j}\big[\sqrt{\mathrm{tr}(\mathbf{K}_j)}\big] + \big(\beta_i^{-\frac{1}{2}} + \beta_j^{-\frac{1}{2}}\big)\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} + \lambda_{i,j}\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} \end{aligned} \tag{4.5}$$

where $\lambda_{i,j} = \min_{h\in\mathcal{H}}\{\mathrm{R}_i(h) + \mathrm{R}_j(h)\}$.

Proof. By Theorem 2.16,

$$\begin{aligned} \mathrm{R}_t(h) \leq & \mathrm{R}_s(h) + \sum_{j=1}^{J} \alpha_j\Big(d_{\mathrm{MMD}}(\mathcal{D}_j, \mathcal{D}^t) + \lambda_j\Big) \\ \leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J} \alpha_j\Big(d_{\mathrm{MMD}}(\mathcal{D}_j, \mathcal{D}^t) + \lambda_j\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} \\ \leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J} \alpha_j\Big(\hat{d}_{\mathrm{MMD}}(\mathbf{X}_j, \mathbf{X}^t) + \frac{2}{\beta_j m}\mathbb{E}_{\mathbf{X}_j}\Big[\sqrt{\mathrm{tr}(\mathbf{K}_j)}\Big] + \frac{2}{m}\mathbb{E}_{\mathbf{X}^t}\Big[\sqrt{\mathrm{tr}(\mathbf{K}^t)}\Big] + \lambda_j\Big) \\ & + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} \\ \leq & \hat{\mathrm{R}}_s(h) + \sum_{i,j=1, i\neq j}^{J+1} (\alpha_i + \alpha_j)\Big(\hat{d}_{\mathrm{MMD}}(\mathbf{X}_i, \mathbf{X}_j) + \frac{2}{\beta_i m}\mathbb{E}_{\mathbf{X}_i}\Big[\sqrt{\mathrm{tr}(\mathbf{K}_i)}\Big] \\ & + \frac{2}{\beta_j m}\mathbb{E}_{\mathbf{X}_j}\Big[\sqrt{\mathrm{tr}(\mathbf{K}_j)}\Big] + \big(\beta_i^{-\frac{1}{2}} + \beta_j^{-\frac{1}{2}}\big)\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} + \lambda_{i,j}\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} \end{aligned} \tag{4.6}$$

This completes the proof. □

Inspired by the multi-class classification, the generalisation bounds of "one-vs-one" domain have been derived. Intuitively, there should be "one-vs-rest" generalisation bounds. Recall the source-combine generalisation bounds in Theorem 2.18, which is derived from the risk of target domain vs the risk of the rest (combined source domains). Hence on the basis of Theorem 2.18, a new "one-vs-rest" generalisation bound can be derived as the following theorem.

Theorem 4.3 (One-vs-rest $\mathcal{H}\Delta\mathcal{H}$ multi-source bound)**.** *Let $\mathcal{H}$ be a hypothesis space of VC dimension* $\mathrm{VC}(\mathcal{H})$*. Let $\mathbf{X}_j$ be labelled samples of size $\beta_j m$ drawn from distribution $\mathcal{D}_j$ with weight α_j for $j \in \{1, \ldots, J+1\}$, where $\alpha_{J+1} = 1$, and $\beta_{J+1} = 1$, let $\mathbf{X}^t = \mathbf{X}_{J+1}$, $\mathbf{X}^s = [\mathbf{X}_1, \ldots, \mathbf{X}_J]$, then for any $\delta \in (0,1)$, with the probability at least $1-\delta$, for every $h \in \mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1} \alpha_j \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}_i, \mathbf{X}_{\alpha\backslash j}) + \lambda_j\big) + \mathrm{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right) \tag{4.7}$$

where $\lambda_j = \min_{h\in\mathcal{H}}\{\mathrm{R}_j(h) + \mathrm{R}_{\alpha\backslash j}(h)\}$, and $\mathrm{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right)$ is the complexity term.

Proof. Let $h_j^* = \arg\min_{h\in\mathcal{H}}\{\mathrm{R}_i(h) + \mathrm{R}_j(h)\}$. Then

$$\begin{aligned}
&|\mathrm{R}_s(h) - \mathrm{R}_t(h)| \\
&\quad = \Big|\sum_{j=1}^{J} \alpha_j \mathrm{R}_j(h) - \mathrm{R}_t(h)\Big| \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \big|\mathrm{R}_{\alpha\backslash j}(h) - \mathrm{R}_j(h)\big| \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \big(\big|\mathrm{R}_{\alpha\backslash j}(h) - \mathrm{R}_{\alpha\backslash j}(h, h_j^*)\big| + \big|\mathrm{R}_{\alpha\backslash j}(h, h_j^*) - \mathrm{R}_j(h, h_j^*)\big| + \big|\mathrm{R}_j(h, h_j^*) - \mathrm{R}_j(h)\big|\big) \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \big(\mathrm{R}_{\alpha\backslash j}(h_j^*) + \big|\mathrm{R}_{\alpha\backslash j}(h, h_j^*) - \mathrm{R}_j(h, h_j^*)\big| + \mathrm{R}_j(h_j^*)\big) \\
&\quad \leq \sum_{j=1}^{J+1} \alpha_j \big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_{\alpha\backslash j}, \mathcal{D}_j) + \lambda_j\big).
\end{aligned} \tag{4.8}$$

The remainder of proof is almost identical to the proof in Theorem 4.1. □

Similar to Theorem 4.1, Theorem 4.3 can also be proved by using the theoretical results in Theorem 2.18:

$$\begin{aligned}\mathrm{R}_t(h) \leq & \hat{\mathrm{R}}_s(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}^t, \mathbf{X}_j^s) + \lambda_{J+1} + \mathrm{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right) \\ \leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1}\alpha_j\big(\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathbf{X}_i, \mathbf{X}_j) + \lambda_j\big) + \left(\mathrm{O}\sqrt{\frac{\mathrm{VC}(\mathcal{H})}{m}}\right)\end{aligned} \tag{4.9}$$

Corollary 4.3.1. *"One vs Rest" bounds are tighter than "one vs one" bound.*

Proof.

$$\begin{aligned}\sum_{j=1}^{J+1}\alpha_j\big|\mathrm{R}_{\alpha\backslash j}(h) - \mathrm{R}_j(h)\big| = & \sum_{j=1}^{J+1}\alpha_j\Big|\sum_{i=1, i\neq j}^{J}\mathrm{R}_i(h) - \mathrm{R}_j(h)\Big| \\ \leq & \sum_{j=1}^{J+1}\alpha_j\sum_{i=1, i\neq j}^{J+1}\big|\mathrm{R}_i(h) - \mathrm{R}_j(h)\big|\end{aligned} \tag{4.10}$$

This completes the proof. □

Same as the "one vs one" bounds, which can derive two generalisation bounds under the frameworks of VC-dimension and Rademacher complexity, respectively, an MMD-based "one-vs-rest" bound for multi-source DA can be derived, which gives the following lemma.

Lemma 4.4 (One-vs-rest MMD multi-source bound)**.** *Let $\mathcal{H}$ be a hypothesis space. Let $\mathbf{X}_j$ be labelled samples of size $\beta_j m$ drawn from distribution $\mathcal{D}_j$ with weight α_j for $j \in \{1, \ldots, J+1\}$, where $\alpha_{J+1} = 1$, and $\beta_{J+1} = 1$, let $\mathbf{X}^t = \mathbf{X}_{J+1}$, $\mathbf{X}^s = [\mathbf{X}_1, \ldots, \mathbf{X}_J]$, and $\mathbf{K}_j$ be the kernel matrix of $\mathbf{X}_j$ with kernel function $k(\cdot, \cdot)$ then for any $\delta \in (0, 1)$, with the probability at least $1 - \delta$, for every $h \in \mathcal{H}$:*

$$\begin{aligned}\mathrm{R}_t(h) \leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1}\alpha_j\Big(\hat{d}_{\mathrm{MMD}}(\mathbf{X}_{\alpha\backslash j}, \mathbf{X}_j) + \frac{2}{(2-\beta_j)m}\mathbb{E}_{\mathbf{X}_{\alpha\backslash j}}\left[\sqrt{\mathrm{tr}(\mathbf{K}_{\alpha\backslash j})}\right] + \\ & \frac{2}{\beta_j m}\mathbb{E}_{\mathbf{X}_j}\left[\sqrt{\mathrm{tr}(\mathbf{K}_j)}\right] + \big((2-\beta_j)^{-\frac{1}{2}} + \beta_j^{-\frac{1}{2}}\big)\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} + \lambda_j\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}}\end{aligned} \tag{4.11}$$

where $\lambda_j = \min_{h\in\mathcal{H}}\{\mathrm{R}_{\alpha\backslash j}(h) + \mathrm{R}_j(h)\}$.

Proof. By Theorem 2.16,

$$\begin{aligned}
\mathrm{R}_t(h) \leq & \mathrm{R}_s(h) + \Big(d_{\mathrm{MMD}}(\mathcal{D}^s, \mathcal{D}^t) + \lambda_{J+1}\Big) \\
\leq & \hat{\mathrm{R}}_s(h) + \Big(d_{\mathrm{MMD}}(\mathcal{D}^s, \mathcal{D}^t)\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} \\
\leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J}\alpha_j\Big(d_{\mathrm{MMD}}(\mathcal{D}_j, \mathcal{D}_{\alpha\backslash j}) + \lambda_j\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} \\
\leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1}\alpha_j\Big(\hat{d}_{\mathrm{MMD}}(\mathbf{X}_{\alpha\backslash j}, \mathbf{X}_j) + \frac{2}{(2-\beta_j)m}\mathbb{E}_{\mathbf{X}_{\alpha\backslash j}}\Big[\sqrt{\mathrm{tr}(\mathbf{K}_{\alpha\backslash j})}\Big] + \\
& \frac{2}{\beta_j m}\mathbb{E}_{\mathbf{X}_j}\Big[\sqrt{\mathrm{tr}(\mathbf{K}_j)}\Big] + \big((2-\beta_j)^{-\frac{1}{2}} + \beta_j^{-\frac{1}{2}}\big)\sqrt{\frac{\ln\frac{2}{\delta}}{2m}} + \lambda_j\Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}}
\end{aligned} \tag{4.12}$$

This completes the proof. □

4.2 Dependence Based Multi-Domain Learning Theory

4.2.1 Hilbert-Schmidt Independence Criterion (HSIC)

Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005) is a non-parametric criterion for measuring the statistical dependence between two sets $\mathbf{X} = \{\mathbf{x}_i\}$ and $\mathbf{Y} = \{\mathbf{y}_i\}$, both with n samples. HSIC is zero if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent, i.e. $\mathbb{P}_{\mathbf{X},\mathbf{Y}} = \mathbb{P}_{\mathbf{X}}\mathbb{P}_{\mathbf{Y}}$. A larger HSIC value suggests stronger dependence. The empirical HSIC between $\mathbf{X}$ and $\mathbf{Y}$, $\rho_h(\mathbf{X}, \mathbf{Y})$ (Gretton et al., 2005), is given by

$$\rho_h(\mathbf{X}, \mathbf{Y}) = \frac{1}{(m-1)^2}\mathrm{tr}(\mathbf{KHLH}), \tag{4.13}$$

where $\mathbf{K}, \mathbf{H}, \mathbf{L} \in \mathbb{R}^{n\times n}$, $\mathbf{K}_{i,j} := k_x(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{L}_{i,j} := k_y(\mathbf{y}_i, \mathbf{y}_j)$, $k_x(\cdot)$, $k_y(\cdot)$ are two kernel functions, $\mathbf{H} = \mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$ is a centring matrix, $\mathbf{I}$ is an identity matrix.

The domain dependence of the data can be computed via $\rho_h(\mathbf{X}, \mathbf{C})$, using the domain covariate matrix $\mathbf{C}$ defined above. Now we show when using *one-hot encoding* and a *linear kernel* for $\mathbf{C}$ in HSIC, we can derive an equivalence between HSIC and MMD, as shown in the

following lemma. Based on this lemma, we can derive generalisation bounds for HSIC-based TL and formulate our new framework.

Lemma 4.5. *Let* $\mathbf{X}^s$, $\mathbf{X}^t$ *be the source and target sample sets with size* m_s *and* m_t, *respectively, and* $\mathbf{c}_0 \in \mathbb{R}^m$ *be a degenerated one-hot domain covariate vector, i.e.* $\mathbf{c}_0 = [\overbrace{0\cdots0}^{m_s}\overbrace{1\cdots1}^{m_t}]$, *where* $m = m_s + m_t$, *then* $\mathrm{HSIC}(\mathbf{X}, \mathbf{c}_0)$ *is proportional to* $\mathrm{MMD}(\mathbf{X}^s, \mathbf{X}^t)$, *where* $\mathbf{X} = [\mathbf{X}^s, \mathbf{X}^t]$, *and linear kernel is used for* $\mathbf{c}_0$ *in HSIC.*

Proof. By Eq. 2.19, the empirical MMD between $\mathbf{X}^s$ and $\mathbf{X}^t$ is

$$\mathrm{MMD}(\mathbf{X}^s, \mathbf{X}^t) = \left\| \frac{1}{m_s} \sum_{i=1}^{m_s} \mathbf{x}_i^s - \frac{1}{m_t} \sum_{i=1}^{m_t} \mathbf{x}_i^t \right\|_{\mathcal{H}_k}^2, \tag{4.14}$$

where $\mathcal{H}_k$ denotes a reproducing kernel Hilbert space (RKHS). The empirical MMD can be computed via $\mathrm{tr}(\mathbf{KL}')$ (Pan et al., 2011), where

$$\mathbf{K} \in \mathbb{R}^{m \times m} = k([\mathbf{X}^s, \mathbf{X}^t], [\mathbf{X}^s, \mathbf{X}^t]) = \begin{bmatrix} \mathbf{K}^{s,s} & \mathbf{K}^{s,t} \\ \mathbf{K}^{t,s} & \mathbf{K}^{t,t} \end{bmatrix}, \tag{4.15}$$

$\mathbf{X} = [\mathbf{X}^s, \mathbf{X}^t] \in \mathbb{R}^{d \times n}$, and $\mathbf{L}' \in \mathbb{R}^{m \times m}$ is defined as

$$\mathbf{L'}_{ij} = \begin{cases} \frac{1}{{m_s}^2} & \text{if} \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^s, \\ \frac{1}{{m_t}^2} & \text{if} \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^t, \\ -\frac{1}{m_s m_t} & \text{otherwise.} \end{cases} \tag{4.16}$$

By Eq. (4.13), $\rho_h(\mathbf{X}, \mathbf{d}_0) = \mathrm{tr}(\mathbf{KHLH})/(n-1)^2$, where $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$ is exactly the same kernel matrix as in the MMD, and $\mathbf{L} = \mathbf{d}_0^\top \mathbf{d}_0$, i.e. $\mathbf{L}_{i,j} = 1$, if $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^t$, and otherwise $\mathbf{L}_{i,j} = 0$. Let $\hat{\mathbf{L}} = \mathbf{HLH}$, resulting

$$\hat{\mathbf{L}}_{ij} = \begin{cases} \frac{{m_t}^2}{m^2} & \text{if} \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^s, \\ \frac{{m_s}^2}{m^2} & \text{if} \quad \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^t, \\ -\frac{m_s m_t}{m^2} & \text{otherwise.} \end{cases} \tag{4.17}$$

By comparing Eq. (4.16) and Eq. (4.17), we have

$$\mathrm{MMD}(\mathbf{X}^s, \mathbf{X}^t) = u\rho_h(\mathbf{X}, \mathbf{c}_0), \tag{4.18}$$

where $u = \frac{m^2(m-1)^2}{(m_s m_t)^2}$, which is a constant for fixed samples in a learning task. This completes the proof. □

By Lemma 4.5, MMD can be viewed as a special case of HSIC, i.e. when there are only two discrete domains and encoded as one-hot covariates, and linear kernel is used for domain covariates when computing HSIC. This is also the theoretical evidence for proving that TCA (Pan et al., 2011) is a special case of MIDA (Yan et al., 2018).

4.2.2 HSIC-Based Generalisation bound

For the multi-source setting, we define a domain j as the samples $\mathbf{X}_j$ drawn from a distribution D_j on the inputs $\mathcal{X}$ and a labelling function $f_j : \mathcal{X} \to \{0, 1\}$. A hypothesis $f \in \mathcal{H}$ is a function $f : \mathcal{X} \to \{0, 1\}$. We consider a classifier f trained on a total of m_s samples $\mathbf{X}^s$ that drawing from J ($J \geq 1$) distinct source domains with a domain weight vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_J]$, where $\sum_{j=1}^{J} \alpha_j = 1$, and derive the bounds on its generalisation performance on a target domain, i.e. $\mathrm{R}_t(f)$ or $\mathrm{R}_t(f, f_t)$.

Theorem 4.6 (HSIC Multi-source Bound)**.** *Let $\mathcal{H}$ be a hypothesis space, $\alpha_{J+1} = 1$, and $\mathcal{D}_{J+1} = \mathcal{D}^t$, for $j \in \{1, \ldots, J\}$, let $\mathbf{X}_j$ be labelled samples of size n_j drawn from D_j with domain weight α_j and labelled by function f_j. Let $\mathbf{C}$ be a one-hot domain covariate matrix, then for $h \in \mathcal{H}$:*

$$\mathrm{R}_t(h) \leq \hat{\mathrm{R}}_s(h) + \rho_h(\mathbf{X}, \mathbf{CU}) + \Omega + \hat{\Re}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}}, \tag{4.19}$$

where

$$\begin{aligned}\Omega = \sum_{j=1}^{J+1} \alpha_j \Big(& \hat{d}_{\mathrm{MMD}}(\mathbf{X}_{\alpha\setminus j}, \mathbf{X}_j) + \frac{2}{(2-\beta_j)m} \mathbb{E}_{\mathbf{X}_{\alpha\setminus j}} \left[\sqrt{\mathrm{tr}(\mathbf{K}_{\alpha\setminus j})} \right] \\ & + \frac{2}{\beta_j m} \mathbb{E}_{\mathbf{X}_j} \left[\sqrt{\mathrm{tr}(\mathbf{K}_j)} \right] + \left((2-\beta_j)^{-\frac{1}{2}} + \beta_j^{-\frac{1}{2}} \right) \sqrt{\frac{\ln \frac{2}{\delta}}{2m}} + \lambda_j \Big),\end{aligned}$$

and $\hat{\mathrm{R}}_s(f)$ *is the empirical risk of* f *on the source data,* $\mathbf{U} = \mathrm{diag}(\mathbf{u})$, $\mathrm{diag}(\cdot)$ *is the diagonal function,* $\mathbf{u} \in \mathbb{R}^{2m}$ *is a vector,* $u_i = 4\alpha_i m^2(2m-1)^2/(m_j^2(2m-m_j)^2)$, *if* $\mathbf{x}_i \in \mathbf{X}_j$, $i = 1, \ldots, 2m$.

Proof. By Lemma 4.4)

$$
\begin{aligned}
\mathrm{R}_t(h) \leq & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1} \alpha_j \Big(\hat{d}_{\mathrm{MMD}}(\mathbf{X}_{\alpha\setminus j}, \mathbf{X}_j) + \frac{2}{(2-\beta_j)m} \mathbb{E}_{\mathbf{X}_{\alpha\setminus j}} \Big[\sqrt{\mathrm{tr}(\mathbf{K}_{\alpha\setminus j})} \Big] + \\
& \frac{2}{\beta_j m} \mathbb{E}_{\mathbf{X}_j} \Big[\sqrt{\mathrm{tr}(\mathbf{K}_j)} \Big] + \big((2-\beta_j)^{-\frac{1}{2}} + \beta_j^{-\frac{1}{2}} \big) \sqrt{\frac{\ln \frac{2}{\delta}}{2m}} + \lambda_j \Big) + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}} \\
= & \hat{\mathrm{R}}_s(h) + \sum_{j=1}^{J+1} \alpha_j \hat{d}_{\mathrm{MMD}}(\mathbf{X}_{\alpha\setminus j}, \mathbf{X}_j) + \Omega + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}} \qquad (4.20) \\
= & \hat{\mathrm{R}}_s(f) + \sum_{j=1}^{J+1} u_j \mathrm{tr}(\mathbf{K}\mathbf{H}\mathbf{L}_j\mathbf{H}) + \Omega + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}} \quad \text{(Lemma 4.5)} \\
= & \hat{\mathrm{R}}_s(h) + \rho_h(\mathbf{X}, \mathbf{C}\mathbf{U}) + \Omega + \hat{\mathfrak{R}}_{\mathbf{X}^s}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}},
\end{aligned}
$$

where $\mathbf{L}_j = \mathbf{d}_j^\top \mathbf{d}_j$, $\mathbf{d}_j \in \mathbb{R}^m$ is the jth row of $\mathbf{C}$, e.g. $\mathbf{c}_1 = [\overbrace{1 \cdots 1}^{\beta_1 m} \overbrace{0 \cdots 0}^{(2-\beta_1)m}]$. This completes the proof. □

Mathematically, the generalisation bound in Theorem 4.6 is equivalent to the one in Lemma 4.4, which is also shown in Eq. 4.20 of the proof. However, HSIC-based bounds have nice properties of higher flexibility and potential for extensions:

- Multiple domain covariates, e.g. subjects' age group, gender, and handiness in neuroimaging analysis, etc.
- Continuous values as covariates, e.g. subjects' age, IQ score, etc.
- Incorporating prior knowledge for covariates, i.e. using other kernel functions, such as Gaussian RBF kernel, instead of linear kernel for computing HSIC.
- Can be estimated in linear time (Jitkrittum et al., 2017; Yokoi et al., 2018; Zhang et al., 2018b) without requiring source target sample sizes to be the same like estimating

linear-time MMD .

4.3 Summary

In this chapter, two multi-source domain adaptation divergence measuring strategies: one-vs-one and one-vs-rest, are developed from Theorem 2.17 and 2.18, respectively. With the two strategies, five generalisation bounds are derived and summarised in Fig. 4.1. Theoretical analysis indicates that bounds with one-vs-rest strategy are tighter than the ones with one-vs-one. In total, there are four generalisation bound introduced for the multi-source domain learning problem, and their relationships can be summarised as the two following inequalities:

$$\underbrace{|\mathrm{A}-\mathrm{BC}|}_{\text{“Source Combine” (Theorem 2.18)}} \leq \underbrace{|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|}_{\text{“One vs Target” (Theorem 2.17)}} \leq \underbrace{|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|+|\mathrm{B}-\mathrm{C}|}_{\text{“One vs One” (Theorem 4.1, 4.2)}},$$

and

$$\underbrace{|\mathrm{A}-\mathrm{BC}|}_{\text{“Source Combine” (Theorem 2.18)}} \leq \underbrace{|\mathrm{A}-\mathrm{BC}|+|\mathrm{B}-\mathrm{AC}|+|\mathrm{C}-\mathrm{AB}|}_{\text{“One vs Rest” (Theorem 4.3,4.4, 4.6)}} \leq \underbrace{|\mathrm{A}-\mathrm{B}|+|\mathrm{A}-\mathrm{C}|+|\mathrm{B}-\mathrm{C}|}_{\text{“One vs One” (Theorem 4.1, 4.2)}}.$$

Moreover, the statistical dependence based bound derived from one-vs-rest MMD bound has higher flexibility. The effectiveness of Theorem 4.6 in multi-source domain adaptation will be evaluated in the next chapter, which is the recommended generalisation bound and one of the main contributions of this thesis.

Chapter 5

Covariate-Independence Regularisation for Unsupervised Multi-Source Adaptation

The previous chapter derived a statistical dependence based generalisation bound for multi-domain learning. Motivated by the theoretical study, this chapter proposes a machine learning framework, Covariate-Independence Regularisation (CoIR), for unsupervised multi-source domain adaptation. Specifically, CoIR simultaneously minimises the empirical risk and the statistical dependence on the domain covariates, to reduce the theoretical generalisation error bound. The CoIR framework will be applied to tackle not only the challenge of learning robust decoding models for fMRI data across different cognitive experiments and subjects, but also unsupervised multi-source domain adaptation of textual sentiment analysis and visual object recognition.

5.1 Introduction

The growth of public neuroimaging data from multiple sites, e.g. the OpenNeuro Gorgolewski et al. (2017), have many similar brain conditions across different cognitive experiments. This enables domain adaptation studies to leverage the power of overlapping labels across domains. Furthermore, it may potentially offer interpretation/insights from the domain shift

perspective for neuroscientists.

Though domain adaptation techniques have been widely applied in computer vision (CV) or natural language processing (NLP) tasks, brain condition decoding presents domain adaptation challenges different from those in CV/NLP. fMRI data are generated by brain signals, which are not natural images that human visual system has adapted to interpret. Consequently, fMRI analysis relies heavily on statistics. Furthermore, cognitive stimuli are implemented varying across experiments. Even the same information can be encoded as different patterns of activity by different brains (Chen et al., 2015). Hence, each subject can be considered as a unique learning task to extract subject-specific features (Rao et al., 2013) in fMRI studies. Additionally, as mentioned before, fMRI data are noisy. Therefore, for domain adaptation in brain condition decoding, it can be beneficial to take more domain information, such as experiment designs and subjects, into account to learn a robust model.

Recently, the *maximum independence domain adaptation* (MIDA) (Yan et al., 2018) introduces a new domain dependence minimisation approach to domain adaptation. It learns common, cross-domain features by minimising statistical dependence on auxiliary domain information, as measured by the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005). This inspired us to encode different experiment designs and subjects as auxiliary domain covariates for domain adaptation in brain condition decoding.

In this chapter, we propose a *Covariate Independence Regularisation* (CoIR) framework for domain adaptation in brain condition decoding. The contributions are threefold: **(1)** We discover the relationship between HSIC and maximum mean discrepancy (MMD) and derive two HSIC-based generalisation bound for multi-source domain adaptation. The theoretical studies enable the formulation of the CoIR framework that simultaneously minimises the empirical prediction risk and the dependence on domain side information. **(2)** Under this framework, we construct a simplified HSIC and incorporate the hinge loss that can take unlabelled samples into account following the Manifold Regularisation framework formulation (Belkin et al., 2006). This gives us the CoIR_{SVM} algorithm. **(3)** We construct new homogeneous brain decoding domain adaptation tasks by identifying datasets with homogeneous brain conditions from public repositories. Experiments on these tasks show the superior performance of CoIR over competing methods.

5.2 Methodology

5.2.1 Multi-Source Domain Adaptation View of Brain Decoding

In a cognitive experiment, each subject is presented a set of stimuli (conditions) designed by neuroscientists. An experiment typically features one or a few (if repeated) samples per condition per subject. We consider a target dataset (experiment) have both labelled and unlabelled samples, and there are labelled samples with the homogeneous brain conditions that acquired from one or more source experiments, where the experiment designs are different. The objective is to predict the human brain conditions of *unlabelled target samples*.

The target cognitive experiment has m_t unlabelled fMRI data samples $\mathbf{X}^t \in \mathbb{R}^{d\times m_t}$, where d is the number of fMRI features, e.g. voxels. The learning task is to classify the samples into C brain conditions (classes), where C = 2 in this chapter, i.e. binary classification.

The source consists of data from one or more cognitive experiments with m_s labelled samples $\mathbf{X}^s \in \mathbb{R}^{d\times m_s}$ in total. The learning task has the same C brain conditions to classify as the target domain.

Domain covariate encoding. Denote the target and source data jointly as $\mathbf{X} = [\mathbf{X}^s, \mathbf{X}^t] \in \mathbb{R}^{d\times m}$, $m = m_s + m_t$. Each fMRI sample $\mathbf{x}_i$ $(i = 1, \cdots, m)$ is collected with a particular experiment implementation j from a particular subject k, where $j = 1, \cdots, p$ and $k = 1, \cdots, q$, i.e. there are p unique experiment implementations and q unique subjects. These are the *domain covariates* to be utilized in our domain adaptation method. We use a simple *one-hot-encoding* strategy to encode such domain covariates. Specifically, we construct a one-hot *experiment implementation* covariate matrix $\mathbf{E} \in \mathbb{R}^{m\times p}$, where its (i, j)th element $e_{i,j} = 1$ if $\mathbf{x}_i$ is collected from experiment j and $e_{i,j} = 0$ otherwise. Similarly, we construct a one-hot *subject* covariate matrix $\mathbf{S} \in \mathbb{R}^{m\times q}$, where $s_{i,k} = 1$ if $\mathbf{x}_i$ is from subject k and $s_{i,k} = 0$ otherwise. We then obtain the auxiliary domain covariate matrix $\mathbf{C} \in \mathbb{R}^{\hat{d}\times m}$ by concatenating $\mathbf{E}^\top$ and $\mathbf{S}^\top$, where $\hat{d} = p + q$.

Multi-source view of brain decoding. As mentioned in Sec. 5.1, each cognitive experiment can be designed differently, and each subject can encode the stimuli differently, i.e. $\mathbb{P}(\mathbf{X}|E_i) \neq \mathbb{P}(\mathbf{X}|E_j)$, and $\mathbb{P}(\mathbf{X}|S_i) \neq \mathbb{P}(\mathbf{X}|S_j)$, where E and S denote an experiment and a subject, respectively. Traditional domain adaptation methods consider two different datasets

(experiments in brain decoding) as different domains. If we also consider each subject as a domain as in Rao et al. (2013), then each unique experiment-subject combination is a domain, i.e. brain decoding domain adaptation tasks is essentially a multi-source transfer problem. Therefore, in the following, we study HSIC in the multi-source domain adaptation setting.

5.2.2 The Covariate-Independence Regularisation Framework

Our ultimate goal is to learn a classifier for the unlabelled target data. From Theorem 4.6, the bound of $\mathrm{R}_t(f)$ can be decreased by simultaneously minimising **1)** the empirical error on labelled data, **2)** the dependence on domain covariates. This observation enables us to propose a new Covariate Dependence Regularisation (CoIR) learning framework that optimises these two objectives, and **3)** model complexity. Here, we follow the Manifold Regularisation framework (Belkin et al., 2006) that can take unlabelled samples into account and formulate CoIR as

$$\min_{f} \quad \underbrace{\mathcal{L}(f(\mathbf{X}^l), \mathbf{Y})}_{\text{Empirical Prediction Risk}} + \underbrace{\mu \|f\|_K^2}_{\text{Tikhonov Regularisation}} + \underbrace{\lambda \rho_h(f(\mathbf{X}), \mathbf{C})}_{\text{Domain Covariate Dependence Regularisation}}, \quad (5.1)$$

where $\mu, \lambda \geq 0$ are hyper-parameters, $\mathbf{X}^l \in \mathbb{R}^{d \times \tilde{m}}$ denotes all labelled samples, and $\mathbf{X}^l = \mathbf{X}^s$ in the unsupervised domain adaptation setting, $f(\cdot)$ is the decision function of a classifier, $\|f\|_K^2$ is the Tikhonov regularisation term, and $\mathbf{Y}$ denotes training labels. For each term in CoIR framework, $\mathcal{L}(f(\mathbf{X}^l), \mathbf{Y})$ minimises the empirical risk, $\|f\|_K^2$ minimises the model complexity, and $\rho_h(f(\mathbf{X}), \mathbf{C})$ minimises the domain dependence in the label decision space.

Interpretations: The classifiers of CoIR can be viewed as a feature mapping, which project the input features to one-dimensional output space, where samples of all domains are in the same distribution. It can also be viewed as regularisation on the hypothesis space $\mathcal{H}$, which further reduces the complexity of $\mathcal{H}$.

5.2.3 Covariate-Independence Regularised SVM and Least Squares

Simplified HSIC with Kernels

In the CoIR framework, we aim to optimise the domain dependence in the decision space. If we view the coefficient vector $\mathbf{w}$ as a *classifier-based feature mapping*, this mapping projects input features to a one-dimensional space (i.e. a line), where the projected values represent the decision scores. Following the principle of dependence minimisation, we aim to learn a domain-independent classifier by minimising the dependence of the decision scores (projected values) on domain covariates, i.e. experiment implementations and subjects. By the Representer Theorem (Schölkopf et al., 2001), we can simplify the HSIC $\rho_h(f(\mathbf{X}), \mathbf{C})$ to the following version

$$\begin{aligned} \rho_{sh}(f(\mathbf{X}), \mathbf{C}) &= \mathrm{tr}((\mathbf{w}^\top \mathbf{K})^\top (\mathbf{w}^\top \mathbf{K})\mathbf{HLH}) \\ &= \mathbf{w}^\top \mathbf{KHLHK}\mathbf{w}, \end{aligned} \tag{5.2}$$

where $\mathbf{L} = \mathbf{C}^\top \mathbf{C}$ (linear kernel) according to Lemma 4.5. The simplified HSIC is constructed directly from classifier outputs so there is no separate feature mapping step as in (Chu et al., 2017; Cao et al., 2018).

Covariate-Independence Regularised SVM ($\mathbf{CoIR_{SVM}}$)

We can plug in any loss function for the first term in CoIR of Eq. (5.1), such as the square loss, logistic loss, or hinge loss. In this chapter, we consider only binary classification with $\mathbf{y} \in \mathbb{R}^{\tilde{m}}$, $y_i \in \{-1, 1\}$, $i = 1, \cdots \tilde{m}$, i.e. decoding C = 2 brain conditions. Here we choose hinge loss, which is robust to binary classification problems, for empirical risk minimisation as in support vector machines (SVMs). We define $f(\mathbf{X}) = \mathbf{w}^\top \phi(\mathbf{X})$, ϕ is a linear or non-linear kernel mapping, $\mathbf{w}$ is a coefficient vector, $y = \mathrm{sgn}(f(\mathbf{x}))$, where $\mathrm{sgn}(\cdot)$ is the sign function that extracts the sign of a real number, i.e. (1 or -1). Using the Representer Theorem again, we have

$$f(\cdot) = \sum_{i=1}^{\tilde{m}} w_i k_x(\cdot, \mathbf{x}_i),$$

Algorithm 1 Covariate Dependence Regularised Support Vector Machine (CoIR_{SVM})

Input: Input data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ (first $\tilde{m}$ samples are labelled), label vector $\mathbf{y} \in \mathbb{R}^{\tilde{m}}$, and domain covariates.
Hyper-parameters: Penalty C, trade-off parameter λ, kernel function $k_x(\cdot,\cdot)$ and corresponding hyper-parameters.
Output: Coefficient vector $\mathbf{w}$.

1: Encode domain covariates into a matrix $\mathbf{C} \in \mathbb{R}^{\hat{d}\times m}$ with one-hot encoding;
2: Construct matrix $\tilde{\mathbf{Y}} \in \mathbb{R}^{\tilde{m}\times m}$, where $\tilde{\mathbf{Y}}_{i,i} = \mathbf{y}_i$, and the rest are zeros, identity matrix $\mathbf{I}$, and centring matrix $\mathbf{H}$;
3: Construct kernel matrices $\mathbf{K} = k_x(\mathbf{X},\mathbf{X})$, $\mathbf{L} = \mathbf{C}^\top\mathbf{C}$;
4: Learn the optimal Lagrange multipliers $\boldsymbol{\alpha}^*$ by solving the QP problem of Eq. (5.8);
5: Compute $\mathbf{w} = (\mathbf{I} + \lambda\mathbf{HLHK})^{-1}\tilde{\mathbf{Y}}^\top\boldsymbol{\alpha}^*$.
6: **return** Coefficient vector $\mathbf{w}$.

and therefore resulting

$$f(\mathbf{x}_j) = \sum_{i=1}^{\tilde{m}} w_i k_x(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{w}^\top k_x(\mathbf{X},\mathbf{x}_j).$$

By incorporating Eq. (5.2) into SVM loss, we formulate the primal objective function of CoIR_{SVM} as

$$\begin{aligned} \min_{\mathbf{w},\xi,b}\ & \frac{1}{2}\mathbf{w}^\top\mathbf{K}\mathbf{w} + C\sum_{i}^{\tilde{m}}\xi_i + \frac{\tilde{\lambda}}{2}\mathbf{w}^\top\mathbf{KHLHK}\mathbf{w}, \\ \text{s.t. }\ & y_i(\mathbf{w}^\top\mathbf{k}_i + b) \geq 1-\xi_i,\ \xi_i \geq 0,\ i = 1,\ldots,\tilde{m}, \end{aligned} \tag{5.3}$$

where $y_i \in \{-1,1\}$, ξ_i is the "slack variable" for the ith sample, b is a bias term, $C > 0$ controls the trade-off between penalty and the margin, and $\tilde{\lambda} = \lambda/(n-1)^2$ controls the significance of simplified HSIC regulariser.

Optimisation of CoIR_{SVM}

To solve Eq. (5.3) effectively, we follow the steps in (Belkin et al., 2006) and reformulate Eq. (5.3) via Lagrange dual. The Problem (5.3) can be rewritten as

$$\begin{aligned} \min_{\mathbf{w},\xi}\ & \frac{1}{2}\mathbf{w}^\top(\mathbf{K} + \tilde{\lambda}\mathbf{KHLHK})\mathbf{w} + C\sum_{i}^{\tilde{m}}\xi_i \\ \text{s.t. }\ & y_i\mathbf{w}^\top\mathbf{k}_i + b \geq 1-\xi_i, \quad i = 1,\ldots,\tilde{m}, \xi_i \geq 0, \quad i = 1,\ldots,\tilde{m}. \end{aligned} \tag{5.4}$$

By defining $\mathbf{G} = \mathbf{K} + \tilde{\lambda}\mathbf{KHLHK}$, the primal problem becomes

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^\top \mathbf{G}\mathbf{w} + C\sum_{i}^{\tilde{m}} \xi_i \\ &\text{s.t. } y_i \mathbf{w}^\top \mathbf{k}_i + b \geq 1 - \xi_i, \quad i = 1, \ldots, \tilde{m}, \xi_i \geq 0, \quad i = 1, \ldots, \tilde{m}. \end{aligned} \tag{5.5}$$

We introduce two sets of Lagrange multipliers $\alpha_i \geq 0$ and $\beta_i \geq 0$, then the Lagrangian of the primal problem is as the following:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\mathbf{w}^\top \mathbf{G}\mathbf{w} + C\sum_{i=1}^{\tilde{m}} \xi_i - \sum_{i=1}^{\tilde{m}} \beta_i \xi_i - \sum_{i=1}^{\tilde{m}} \alpha_i (y_i(\mathbf{w}^\top \mathbf{k_i} + b) - 1 + \xi_i). \tag{5.6}$$

Computing the gradients w.r.t. b and ξ_i, and let them equal to zero, resulting

$$\frac{\partial L}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{\tilde{m}} \alpha_i y_i = 0,$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Longrightarrow C - \alpha_i - \beta_i = 0 \Longrightarrow 0 \leq \alpha_i \leq C.$$

Then Eq. (5.6) becomes

$$\begin{aligned} L(\mathbf{w}, \alpha) &= \frac{1}{2}\mathbf{w}^\top \mathbf{G}\mathbf{w} - \sum_{i=1}^{\tilde{m}} \alpha_i (\tilde{y}_i \mathbf{w}^\top \mathbf{k_i} - 1) \\ &= \frac{1}{2}\mathbf{w}^\top \mathbf{G}\mathbf{w} - \mathbf{w}^\top \mathbf{K}\tilde{\mathbf{Y}}\boldsymbol{\alpha} + \sum_{i=1}^{n} \alpha_i, \end{aligned} \tag{5.7}$$

where $\tilde{\mathbf{Y}} \in \mathbb{R}^{\tilde{m} \times m}$, $\tilde{\mathbf{Y}}_{i,i} = \mathbf{y}_i$, $i = 1, \ldots \tilde{m}$, and the rest are zeros, by considering the first $\tilde{m}$ samples are labelled. Let $\mathbf{Q} = \tilde{\mathbf{Y}}\mathbf{K}(\mathbf{I} + \tilde{\lambda}\mathbf{HLHK})^{-1}\tilde{\mathbf{Y}}^\top$ and leads to the dual formulation

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{Q}\boldsymbol{\alpha} - \sum_{i=1}^{\tilde{m}} \alpha_i, \\ &\text{s.t. } \sum_{i=1}^{\tilde{m}} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, \tilde{m}. \end{aligned} \tag{5.8}$$

Equation (5.8) is a quadratic programming (QP) problem that can be solved by standard

QP tools. Denoting $\boldsymbol{\alpha}^*$ as the optimal solution to Eq. (5.8), and the gradients w.r.t. $\mathbf{w}$ is

$$\frac{\partial L}{\partial \mathbf{w}} = (\mathbf{K} + \tilde{\lambda}\mathbf{KHLHK})\mathbf{w} - \mathbf{K}\tilde{\mathbf{Y}}\boldsymbol{\alpha}^*.$$

Then the optimal $\mathbf{w}$ can be obtained via

$$\mathbf{w} = (\mathbf{I} + \tilde{\lambda}\mathbf{HLHK})^{-1}\tilde{\mathbf{Y}}\boldsymbol{\alpha}^*.$$

Algorithm 1 is the pseudo-code for CoIR_{SVM}.

Covariate-Independence Regularised Least Squares (CoIR_{LS})

The regularised least squares classifier is a supervised learning algorithm to solve:

$$\min_{f\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}(y_i - f(\mathbf{x}_i))^2 + \mu\|f\|_K^2. \tag{5.9}$$

Using the Representer Theorem again, the objective of least squares algorithm becomes:

$$\arg\min_{\mathbf{w}}(\mathbf{Y} - \mathbf{Kw})^\top(\mathbf{Y} - \mathbf{Kw}) + \frac{\mu}{2}\mathbf{w}^\top\mathbf{Kw} \tag{5.10}$$

Similar to the formulation of CoIR_{SVM}, by incorporating simplified HSIC and considering the unlabelled target domain data, the objective function of CoIR_{LS} can be formulated as

$$\arg\min_{\mathbf{w}}(\mathbf{Y} - \mathbf{JKw})^\top(\mathbf{Y} - \mathbf{JKw}) + \frac{\mu}{2}\mathbf{w}^\top\mathbf{Kw} + \frac{\tilde{\lambda}}{2}\mathbf{w}^\top\mathbf{KHLHKw}, \tag{5.11}$$

where $\mathbf{Y} \in \mathbb{R}^{m\times m}$, $\mathbf{Y}_{i,i} = \mathbf{y}_i$, for $i = 1, \ldots \tilde{m}$, and $\mathbf{J} \in \mathbb{R}^{m\times m}$, $\mathbf{J}_{i,i} = 1$, for $i = 1, \ldots \tilde{m}$. By taking the partial derivative of (5.11) with regard to $\mathbf{w}$ and let it equals to zero, the optimal $\mathbf{w}$ can be obtained via

$$\mathbf{w} = (\mathbf{JK} + \mu\mathbf{I} + \tilde{\lambda}\mathbf{HLHK})^{-1}\mathbf{Y}. \tag{5.12}$$

Algorithm 2 is the pseudo code for CoIR_{LS}.

Algorithm 2 Covariate Dependence Regularised Least Squares (CoIR_{LS})

Input: Input data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ (first $\tilde{m}$ samples are labelled), label vector $\mathbf{y} \in \mathbb{R}^{\tilde{m}}$, and domain covariates.
Hyper-parameters: μ_1, μ_2, kernel function $k_x(\cdot, \cdot)$ and corresponding hyper-parameters.
Output: Coefficient vector $\mathbf{w}$.

1: Encode domain covariates into a matrix $\mathbf{C} \in \mathbb{R}^{\hat{d} \times m}$ with one-hot encoding;
2: Construct matrix $\mathbf{Y} \in \mathbb{R}^{m \times m}$, where $\mathbf{Y}_{i,i} = \mathbf{y}_i$, and the rest are zeros, for $i = 1, \ldots, \tilde{m}$, matrix $\mathbf{J} \in \mathbb{R}^{n \times n}$, where $\mathbf{J}_{i,i} = 1$, and the rest are zeros, for $i = 1, \ldots, \tilde{m}$, identity matrix $\mathbf{I}$, and centring matrix $\mathbf{H}$;
3: Construct kernel matrices $\mathbf{K} = k_x(\mathbf{X}, \mathbf{X})$, $\mathbf{L} = \mathbf{C}^\top \mathbf{C}$;
4: Compute $\mathbf{w} = (\mathbf{JK} + \mu_1 \mathbf{I} + \mu_2 \mathbf{HLHK})^{-1} \mathbf{Y}$.
5: **return** Coefficient vector $\mathbf{w}$.

5.2.4 Analysis

The Role of Unlabelled Samples

The unlabelled target samples have no labels, but they have domain covariates available. Thus, they affect (regularise) the model coefficients $\mathbf{w}$ directly via the simplified HSIC term.

Computational Complexity

The complexity of computing HSIC is $\text{O}(m(d^2 + \tilde{d}^2))$ when linear kernel is used, i.e. $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$, $\mathbf{L} = \mathbf{C}^\top \mathbf{C}$ (Gretton et al., 2005). For the multi-source domain adaptation problems considered in this thesis, $d \gg \tilde{d}$, so the overall computational complexity of HSIC is $\text{O}(md^2)$. e.g. brain decoding, $d \gg m$ and For hinge loss, the complexity of solving the quadratic programming problem for Eq. (5.8) is $\text{O}(m^3)$, which is also the complexity for solving Eq. (5.12) of least square loss. To sum up, for multi-source domain adaptation tasks with high-dimensional features and *smaller sample sizes*, such as brain decoding, the computational complexity of both CoIR_{SVM} and CoIR_{LS} is $\text{O}(md^2)$. For the tasks with *larger sample size*, such as sentiment analysis and object recognition, the computational complexity of both CoIR_{SVM} and CoIR_{LS} is $\text{O}(m^3)$

Connection to Existing Methods

CoIR minimises prediction error and domain dependence simultaneously, and therefore it can also be viewed as combining the virtues from both domain-invariant classifier methods and

domain dependence minimisation mapping. We summarise the relationship between CoIR and related methods as follows.

CoIR vs. ARTL. By Lemma 4.5, ARTL without manifold regularisation and conditional distribution mismatch is equivalent to CoIR with the degenerated domain covariate matrix $\mathbf{C}_0$. However, CoIR can model multiple sources and domain covariates, making it more flexible than ARTL. Moreover, it is easier to extend CoIR to leverage the rich continuous covariates in public neuroimaging dataset, such as subjects' age, IQ, and handiness score. For the same reason, TCA is equivalent to MIDA with $\mathbf{c}_0$.

CoIR vs. MIDA. We can also view CoIR as 1) replacing the label dependence term $\rho_h(\phi(\mathbf{X}), \mathbf{Y})$ in semi-supervised version of MIDA with the prediction loss, and 2) learning a mapping to a *one-dimensional* classification space (i.e. a line) rather than a low-dimensional subspace.

CoIR vs. Multi-task learning (MTL). MTL learns N different hypotheses for predicting unlabelled samples from N tasks. CoIR learns only one hypothesis (homogeneous task) for predicting unlabelled target samples.

5.3 Experiments

5.3.1 Multi-Source Domain Adaptation Tasks and Datasets

Brain Decoding with OpenNeuro (Stop-Signal Tasks) Data

The datasets of five stop-signal cognitive experiments from the public OpenNeuro repository[1] (Gorgolewski et al., 2017) are selected for multi-source adaptation of brain decoding, which are the experiments of A to E summarised in Table 5.1. Each dataset is from an experiment with specific cognitive settings, but under the same paradigm of stop-signal. Subjects from the same accession number (ds×××) are the same and there is no overlapping subject between accession numbers. Specifically, there were 20 unique subjects involved in ds007 (Experiments A, B, and C), and 13 subjects in ds008 (Experiments D and E). There are two brain conditions "*Successful stop*" and "*Unsuccessful stop*" selected from each dataset, with each as a *class*

[1]We used the OpenNeuro data in the legacy format available at `https://legacy.openfmri.org`

and having the same number of samples. Thus, we have binary classification problems that discriminate between brain conditions in an experiment.

Neuroimaging preprocessing. Using the same data processing workflow introduced in Section 3.3.1 (Chapter 3), the neuroimaging samples were preprocessed using FSL (Jenkinson et al., 2012) with the protocol in (Poldrack et al., 2013) to obtain the Z-score statistical parametric maps (SPMs) (Friston et al., 1994, 1998) of size $91 \times 109 \times 91$, which is then reduced to a vector of size $228,546$ by masking the voxels outside of the brain.

Sentiment Analysis with Amazon Review Data

Sentiment analysis is a binary classification task to identify whether the sentiment of a product review is positive or negative. We use the multi-domain Amazon reviews dataset (Blitzer et al., 2007; Chen et al., 2012), which consists of the reviews from four categories of products on Amazon: Books (B), DVDs (D), Electronics (E), and Kitchen appliances (K). For each domain, there are 2,000 examples (1,000 positive and 1,000 negative), and each sample has been processed and represented as 5,000 TF-IDF features.

Visual Object Recognition with Office-Caltech Data

This is a multi-class classification task to recognise the category of object from images. The Office-Caltech dataset (Gong et al., 2012) contains 2533 images of the ten overlapped classes between the Office-31 (Saenko et al., 2010) and Caltech-256 datasets. There are four domains in this dataset: Amazon (A), Caltech (C), DSLR (D), and Webcam (W). In the experiments, ResNet (ResNet50) (He et al., 2016), a Convolutional Neural Network (CNN) pre-trained on the ImageNet dataset (Deng et al., 2009), will be used to extract 1,000 dimensional features for the non-deep methods, and as the backbone network for the deep learning based methods.

5.3.2 Experimental Setup

Baseline

We evaluate CoIR framework against the following methods: two standard source only baselines 1) **SVM** and 2) **Deep Neural Networks (NN)**; six single source domain adaptation

Table 5.1: Information on the OpenNeuro data used. 'Exp' indexes the five cognitive experiments A–E. #AC is the accession number of an OpenNeuro project, where the same group of subjects are used in each project and there is no overlapping subject between projects. #Sub indicate the number of unique subjects for each dataset. Each of the five experiments has two brain conditions to classify, which are "*Successful stop*" and "*Unsuccessful stop*". Each subject in each experiment contributed two positive and two negative brain condition samples, respectively.

Exp	#AC	Exp Description	#Sub
A	ds007	Stop signal with spoken pseudo word naming Xue et al. (2008)	20
B	ds007	Stop signal with spoken letter naming Xue et al. (2008)	20
C	ds007	Stop signal with manual response Xue et al. (2008)	20
D	ds008	Unconditional stop signal Aron et al. (2007)	13
E	ds008	Conditional stop signal Aron et al. (2007)	13

methods (combine and view all source domains as a single domain): 3) **TCA** (Pan et al., 2011), 4) **ARSVM** & 5) **ARRLS** under ARTL (Long et al., 2013a), 6) **MEDA** (Wang et al., 2018) 7) **DANN** (Ganin et al., 2016), 8) **DAN** (Long et al., 2019); and three state-of-the-art multi-source methods: 9) **MIDA** (Yan et al., 2018) , 10) **MFSAN** (Zhu et al., 2019), 11) **M^{3}SDA** (Peng et al., 2019). These baselines can also be categorised as follows:

- **None-deep methods**: SVM, TCA, ARSVM, ARRLS, MEDA, MIDA
- **Deep-learning methods**: NN (MLP or ResNet), DANN, DAN, MFSAN, M^{3}SDA.

For all of the deep neural networks, we use the implementations from the Python package PyKale (Lu et al., 2021) of the PyTorch ecosystem. For each learning task, deep methods will be repeated five times under the same setting with different random seeds and the mean accuracy will be reported. Since the dimension of neuroimaging data is too high for the deep learning methods, PCA will be applied first to reduce the feature dimension to 350 (97% of variance kept). For both brain decoding and sentiment analysis tasks, because the input data has been pre-processed as vectors, and the relationship between features is different from the image pixels in the object recognition task, CNN is not appropriate to be used as backbone feature extractor. Therefore, a three layer fully connected network, i.e. multilayer perceptron (MLP), is used as the backbone for both non-adaptation NN and DA networks.

For brain decoding tasks, both experiments and subjects will be considered as domain covariates for dependence minimisation approaches, i.e. MIDA and CoIR. Figure 5.1 gives

	Exp 1	...	Exp p	Sub 1	...	Sub q
Sample 1	1	0	0	1	0	0
Sample 2	1	0	0	1	0	0
Sample 3	0	1	0	0	1	0
Sample 4	0	1	0	0	1	0
...	...	...	...	...	...	...
Sample m	0	0	1	0	0	1

Figure 5.1: Example of a domain covariate matrix for brain decoding learning task.

an example of the domain covariate matrix $\mathbf{C}$. For other approaches, samples acquired from one cognitive experiment is considered as a domain.

In object recognition, a further study will be to understand the effectiveness of the four multi-domain generalisation bounds, which are "source-combine" (Theorem 2.18), "one vs target" (Theorem 2.17), "one vs one" (Theorem 4.1 and 4.2), and "one vs rest" (Theorem 4.3, 4.4, and 4.6). Three deep learning baselines will be selected for experiments: DAN for "source combine", MFSAN for "one vs target", and M^{3}SDA for "one vs one". Because the divergence metric adopted in M^{3}SDA (k-moment) is different from the one in DAN and MFSAN (MMD), a new variant of M^{3}SDA is developed by replacing k-moment with MMD to make it comparable. Moreover, another variant of M^{3}SDA with HSIC as divergence is also developed to construct a neural network for the "one vs rest" bound. The two new invariants of M^{3}SDA are named as M^{3}SDA$_{\text{MMD}}$ and M^{3}SDA$_{\text{CoIR}}$.

Cross-Validation Strategy

Leave-one-domain-out is the basic cross validation strategy for the experiments. For non-adaptation approaches, the models will be trained on source domains, and then tested on held out target domain data. For domain adaptation approaches, including single- and multi-source adaptation methods, both source and target data are used for training, however the labels of target instances are unknown for the models.

The optimal values of hyper-parameters for all methods determined by the same leave-one-domain-out cross-validation strategy on source domain data only. For non-deep methods,

Table 5.2: Binary brain condition decoding accuracy (%) on the OpenNeuro Stop-signal data with whole brain features obtained by **non-deep** approaches. 'Avg' is the weighted average accuracy (by domain sample sizes) over the five tasks. The abbreviations of the proposed methods are both *italicised and underlined*. The best result for each task is in **bold**, and the second best is underlined.

	Source Only	Source Combine			Multi-Source		
Target	SVM	TCA	ARSVM	ARRLS	MIDA	$CoIR_{SVM}$	$CoIR_{LS}$
A	71.25	76.83	74.00	73.75	73.75	80.00	**83.76**
B	67.11	**79.27**	68.42	67.11	68.42	78.95	77.63
C	62.82	64.10	69.23	66.67	62.82	64.10	**69.23**
D	73.17	76.83	76.83	80.49	79.27	80.49	**85.37**
E	80.49	79.27	85.26	85.27	86.59	85.37	**90.24**
Avg	70.01	74.81	73.74	73.33	72.74	76.94	**80.18**

i.e. TCA, MIDA, ARSVM, ARRLS, MEDA, CoIR_{SVM}, and CoIR_{LS}, the value of hyper-parameter for model coefficients regularisation (e.g. C for CoIR_{SVM} and μ for CoIR_{LS}) is fixed to be 1 at first and then search the optimal values for the importance of domain divergence regularisation (e.g. λ for CoIR_{SVM} and CoIR_{LS}) in the range of [0.001, 0.01, 0.1, 1, 10, 100, 1000]. Then fix the optimal value of hyper-parameter for domain divergence regularisation, and then search the optimal values for model coefficients regularisation also in [0.00, 0.01, 0.1, 1, 10, 100, 1000]. For the dimension reduction methods, i.e. TCA and MIDA, the optimal values of feature dimension will be searched in [128, 256, 350], followed by a kernel SVM with the optimal values are determined by the grid search functions implemented in the Scikit-learn library (Pedregosa et al., 2011). The hyper-parameters for deep neural networks can be determined adaptively by the pipelines in PyKale library.

5.3.3 Results

Brain Decoding

Table 5.2 presents the brain decoding accuracy on whole brain features. Among all DA methods, the proposed CoIR_{LS} obtained the best performance with a 5.37% improvement over the best comparing method (TCA, 74.81%) on average. CoIR_{SVM} is the second best algorithm, with a 3.24% difference behind CoIR_{LS}.

Table 5.3: Binary brain condition decoding accuracy (%) on the OpenNeuro Stop-signal data with features extracted by PCA. 'Avg' is the weighted average accuracy (by domain sample sizes) over the five tasks. The abbreviations of deep-learning methods are *ITALICISED*, and the proposed methods are both *italicised and underlined*. The best result for each task is in **bold**, and the second best is underlined.

	Source Only		Source Combine					Multi-Source			
Target	SVM	*NN*	ARSVM	ARRLS	MEDA	*DANN*	*DAN*	*M³SDA*	*MFSAN*	*CoIR$_{SVM}$*	*CoIR$_{LS}$*
A	72.50	50.07	76.25	80.00	73.75	48.40	49.00	49.87	56.10	**81.25**	80.00
B	63.16	47.94	69.74	75.00	69.74	51.13	49.06	49.61	50.91	73.68	**75.00**
C	66.67	49.75	71.79	**75.64**	64.10	50.06	52.62	53.51	54.81	71.79	**75.64**
D	64.63	48.07	81.71	84.15	81.71	54.20	52.53	54.29	58.44	**85.37**	85.37
E	69.51	48.07	84.15	81.71	87.81	52.47	48.07	53.51	59.48	**87.80**	82.93
Avg	67.29	48.89	76.73	79.30	75.42	51.25	50.26	52.16	55.95	**79.98**	79.79

Table 5.3 reports the decoding accuracy on the fMRI features extracted by PCA. The results obtained by all deep neural networks are below 60%, which shows the difficulty of learning the model parameters with such small sample sizes (less than 400) for deep learning based methods. For the results obtained by the non-deep methods, CoIR classifiers still performed the best. CoIR$_{\text{SVM}}$ obtained the highest weighted-average accuracy 79.98%. However, the improvement is not as significant as on whole brain features. The accuracy of ARSVM, ARRLS, and CoIR$_{\text{SVM}}$ improved significantly on PCA features, the improvement over these comparing methods reduced to 0.68% (vs 79.30% by ARRLS).

In general, for brain decoding tasks, all domain adaptation methods (non-deep) outperformed the none-adaptation SVM, and the improvement is larger on PCA features. By contrast, the improvement of CoIR over the comparing domain adaptation baselines is larger on whole brain features. Furthermore, hundreds of examples are not enough to train deep neural networks for decoding brain conditions.

Sentiment Analysis

Table 5.4 shows the sentiment classification results of four multi-source adaptation tasks across different algorithms. Two main observations can be summarised from the results:

- Multi-source methods outperformed the corresponding single-source (i.e. source combine) methods, i.e.. CoIR$_{\text{SVM}}$ and CoIR$_{\text{LS}}$ vs ARSVM, ARRLS, and MEDA, MIDA vs

Table 5.4: Binary sentiment classification accuracy (%) on the Amazon review data. The four domains are Books (B), DVDs (D), Electronics (E), and Kitchen appliances (K), where each domain contains 2000 product reviews (1000 positive and 1000 negative). 'Avg' is the averaged accuracy over the four tasks. The abbreviations of deep-learning methods are *ITALICISED*, and the proposed methods are both *italicised and underlined*. The best result for each task is in **bold**, and the second best is underlined.

	Source Only		Source Combine						Multi-Source				
Target	SVM	*NN*	TCA	ARSVM	ARRLS	MEDA	*DANN*	*DAN*	MIDA	*M³SDA*	*MFSAN*	*CoIR$_{SVM}$*	*CoIR$_{LS}$*
B	73.35	66.79	74.05	77.25	76.00	71.90	73.20	74.48	74.15	76.06	71.37	79.75	**80.30**
D	76.60	67.80	74.40	79.70	76.90	74.25	74.61	75.97	75.40	76.24	69.89	80.20	**81.70**
E	84.40	71.16	80.15	81.70	85.50	80.30	79.76	81.48	80.50	66.60	75.01	84.65	**85.55**
K	**85.70**	72.84	80.90	85.65	85.00	81.18	82.85	52.86	81.45	79.32	76.36	85.55	85.50
Avg	80.01	69.65	77.38	81.08	80.85	76.99	77.19	78.69	77.88	74.56	73.16	82.54	**83.26**

TCA, M^{3}SDA and MFSAN vs DANN and DAN. On average CoIR$_{\text{LS}}$ obtained the best accuracy (83.26%), and CoIR$_{\text{SVM}}$ is the second best. The improvement over the best comparing baseline is 2.18% (vs 81.08% by ARSVM).

- "Negative" transfer is observed. Only classifier learning based methods CoIR and ARTL outperformed the baseline SVM. This may indicate the domain adaptation regularisation is more effective via regularising classifiers directly. For deep domain adaptation methods, all of them outperformed the non-adaptation deep-learning baseline NN(MLP) but could not outperform SVM.

Visual Object Recognition

Table 5.5 and 5.5 reports the accuracy of object recognition on the Office-Caltech dataset. The deep neural networks outperformed the non-deep methods significantly on this task. The observations for these two tables will be summarised separately as follows:

- For non-deep methods, almost all classifier-based methods outperformed SVM. By contrast, all feature mapping approaches, i.e. TCA and MIDA, could not outperform SVM. CoIR$_{\text{LS}}$ is still the best performing non-deep method. However, the improvement is not significant compared to ARRLS and MEDA.

- All DA neural networks outperformed the fine-tuned ResNet50 (NN), and multi-source approaches, i.e. MFSAN and M^{3}SDA, outperformed single-source approaches DANN

Table 5.5: Ten-class object recognition accuracy (%) on the Office-Caltech dataset obtained by **non-deep** methods. The four domains are Amazon (A): 958 images, Caltech (C): 1123 images, DSLR (D): 157 images, and Webcam (W): 295 images. 'Avg' is the weighted (by domain sample size) average accuracy over the four tasks. The abbreviations of the proposed methods are both *italicised and underlined*. The best result for each task is in **bold**, and the second best is underlined.

	Source Only	Source Combine				Multi-Source		
Target	SVM	TCA	ARSVM	ARRLS	MEDA	MIDA	*CoIR$_{SVM}$*	*CoIR$_{LS}$*
A	80.48	76.93	81.52	82.46	81.73	77.45	78.29	**84.03**
C	68.30	60.64	70.70	**72.84**	72.75	57.17	66.43	72.04
D	91.72	82.17	90.45	**96.82**	92.36	83.44	87.90	93.63
W	82.37	74.24	84.07	87.12	82.03	77.29	74.58	**90.85**
Avg	76.00	69.72	77.58	79.63	78.44	68.81	73.20	**80.10**

and DAN. Better results were obtained by M^{3}SDA$_{\text{MMD}}$ and M^{3}SDA$_{\text{CoIR}}$, the two variants of networks developed on the basis of the baseline M^{3}SDA. It should also be noticed that M^{3}SDA$_{\text{CoIR}}$ not only outperformed all of the baselines here, but also outperformed the results on the same dataset reported in the original paper of M^{3}SDA (Peng et al., 2019), where a much deeper backbone network, ResNet-101, was used.

Hyper-Parameter Sensitivity

The sensitivity of CoIR$_{\text{SVM}}$ and CoIR$_{\text{LS}}$ with linear kernels against their hyper-parameters are evaluated under leave-one-domain-out setting. As shown in Fig. 5.2a and 5.2c, both CoIR$_{\text{SVM}}$ and CoIR$_{\text{LS}}$ are not sensitive to λ, which indicates the stability of covariate-independence regularisation importance. Figure 5.2b shows the sensitivity of CoIR$_{\text{SVM}}$ against $C \in [10^{-4}, 10^{3}]$ when fixing $\lambda = 1$. It can be observed that the accuracy stays stable when $C \geq 0.1$. When $C \in [10^{-4}, 10^{-1}]$ the accuracy is unstable but there is a trend of increase with the growth of C. Since a larger value of C can lead to a smaller SVM classification margin and weaker regularisation on model parameters, which suggests less weight on model parameter regularisation can lead to higher prediction accuracy CoIR$_{\text{SVM}}$. For CoIR$_{\text{LS}}$, it is not sensitive to μ, while larger values ($\geq 10^{3}$) can also lead to changing of performance, as shown in Fig. 5.2d.

Table 5.6: Ten-class object recognition accuracy (%) on the Office-Caltech dataset obtained by **deep neural networks**. The four domains are Amazon (A): 958 images, Caltech (C): 1123 images, DSLR (D): 157 images, and Webcam (W): 295 images. 'Avg' is the weighted (by domain sample size) average accuracy over the four tasks. M^3SDA_{MMD} and M^3SDA_{CoIR} are two new variants developed on the basis of the baseline M^3SDA by replacing the domain divergence metric k-moment with MMD and HSIC, respectively. The best result for each task is in **bold**, and the second best is underlined. The effectiveness of the four multi-domain generalisation bounds can be reflected by: DAN-"source-combine" (Theorem 2.18), MFSAN-"one vs target" (Theorem 2.17), M^3SDA_{MMD}-"one vs one" (Theorem 4.2), and M^3SDA_{CoIR}-"one vs rest" (Theorem 4.6)

	Source Only	Source Combine		Multi-Source			
Target	NN	DANN	DAN	MFSAN	M^3SDA	M^3SDA_{MMD}	M^3SDA_{CoIR}
A	68.01	67.96	86.73	90.50	91.94	93.41	**95.30**
C	66.07	66.17	77.67	87.90	82.37	91.81	**92.90**
D	86.41	86.28	95.21	86.11	94.50	91.01	**98.53**
W	82.43	82.43	92.79	87.55	94.27	91.10	**98.14**
Avg	68.97	69.99	83.95	88.73	87.96	92.28	**94.76**

5.4 Discussion

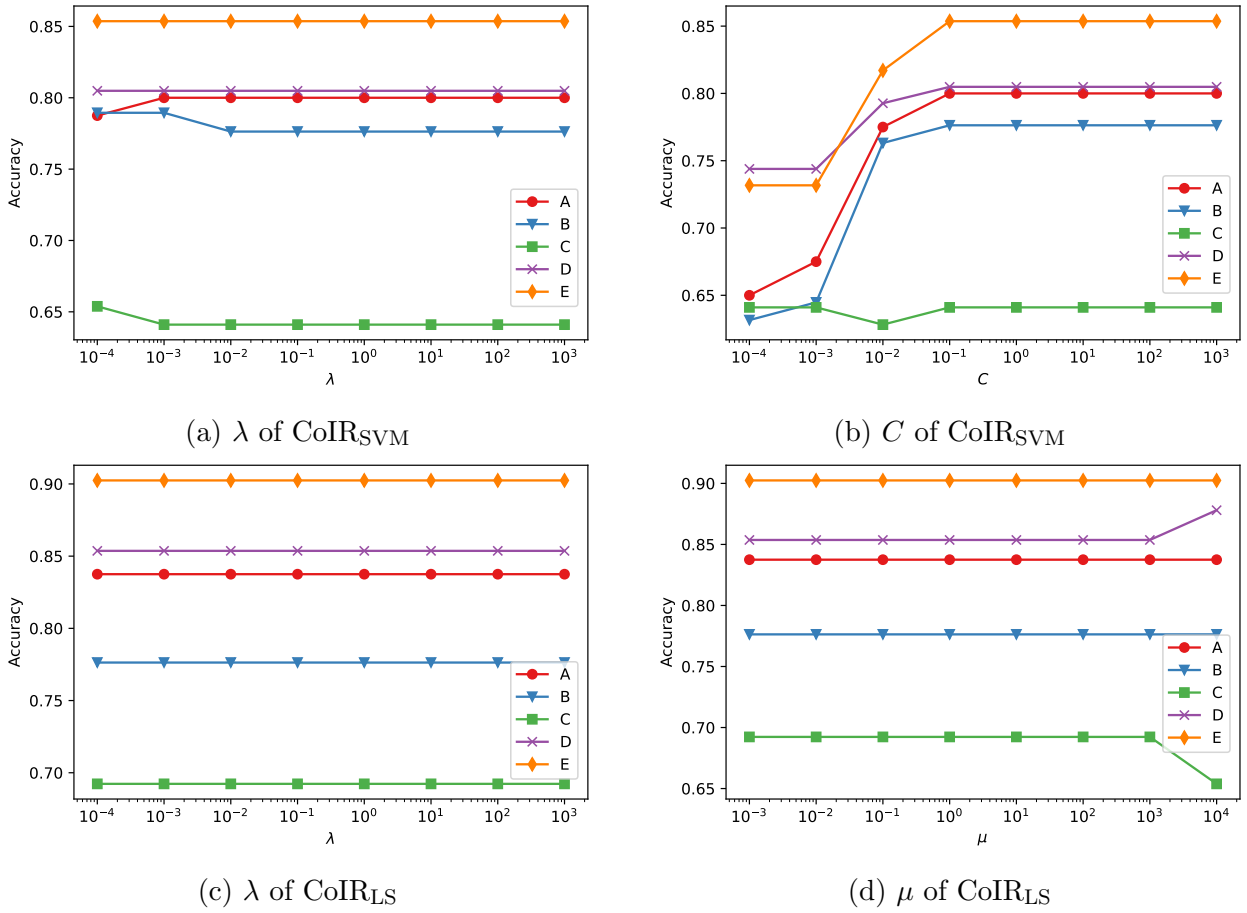
5.4.1 Further Analysis of Results

Effectiveness of Covariate-Independence Regularisation

For the three learning tasks in experiments, all of the best results are obtained by covariate-independence regularised methods, including a deep neural network on visual object recognition tasks. This confirms not only the effectiveness of CoIR, but also the high potential applications in deep learning to explore. In addition, domain adaptation classifiers outperformed their corresponding feature learning methods on average. This suggests the source label information is helpful in learning domain-generic patterns.

Advantage of Neural Networks

Deep neural networks could not outperform non-deep domain adaptation approaches on the brain decoding and textual sentiment analysis. The use PCA and TF-IDF features may be one possible reason to the underperformance of the deep learning methods, and as a result



(a) λ of CoIR_{SVM} (b) C of CoIR_{SVM}

(c) λ of CoIR_{LS} (d) μ of CoIR_{LS}

Figure 5.2: Sensitivity of the classification accuracy with respect to hyper-parameters of CoIR_{SVM} and CoIR_{LS} on the five multi-source brain decoding tasks.

the power of deep neural networks to learn "good" representations (Bengio, 2012) is not fully exploited. The good performance on visual object recognition can be an evidence, where raw images are used as input, and a well-designed and pre-trained backbone network is used.

Link to Generalisation Bounds

According to the experimental results of object recognition, where $\text{M}^3\text{SDA}_{\text{CoIR}}$ outperformed $\text{M}^3\text{SDA}_{\text{MMD}}$, MFSAN, and DAN, the empirical generalisation risk/error (the lower the better) can be summarised as

$$\text{"One vs Rest"} \leq \text{"One vs One"} \leq \text{"One vs Target"} \leq \text{"Source Combine"}.$$

This confirmed the effectiveness of the proposed "one vs rest" (dependence/HSIC-based) generalisation bound, and therefore it is recommanded to apply this bound/regularisation in multi-source problems. However, this inequality is inconsistent with the theoretical results summarised at the end of Chapter 4, where the relationship between the four multi-domain generalisation bounds in terms of tightness (the lower the better) are

$$\text{"Source Combine"} \leq \text{"One vs Target"} \leq \text{"One vs One"},$$

and

$$\text{"Source Combine"} \leq \text{"One vs Rest"} \leq \text{"One vs One"}.$$

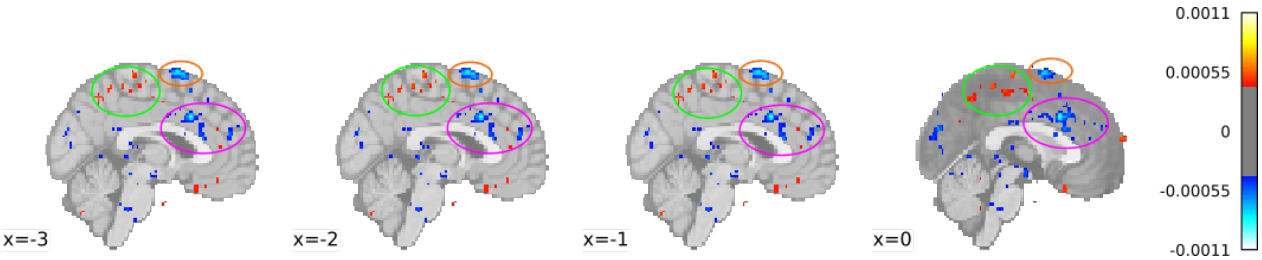
Moreover, in most cases on all of the three learning tasks, single-source (source-combine) methods could not outperform multi-source methods, which is also inconsistent with the theory. This needs to be further studied in the future.

Choice of Kernels

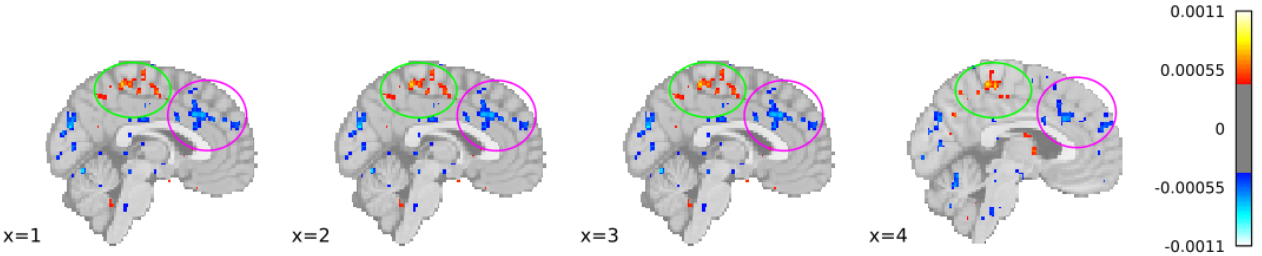
For CoIR on brain decoding tasks, the linear kernel is selected by cross validation and obtained the best results. This indicates the high-dimension neuroimaging features are rich and also benefits the model interpretability. For sentiment analysis and visual object recognition, which have much more train instances and lower feature dimension compared to brain decoding, the RBF kernel is selected and obtained better accuracy. These results are consistent with the suggestion of (Hsu et al., 2003), where a linear kernel is suggested when the feature dimension is very high but the sample size is small.

5.4.2 Model Visualisation and Interpretation

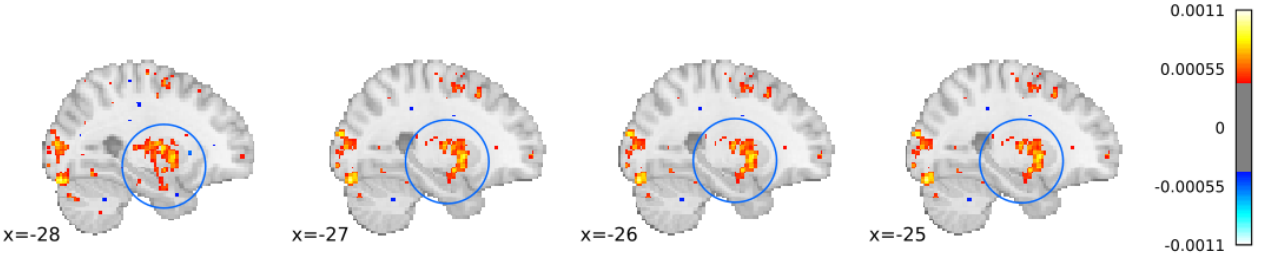
After validating the effectiveness of CoIR, the best performing CoIR_{LS} is re-trained on the five datasets. The top 1% coefficients in magnitude of the obtained model are visualised for interpretation, which are shown in Figure 5.3. Figure 5.3a and 5.3b show the identified activation areas (in cluster) in cerebral cortex, where the cingulate gyrus (marked by pink circles), and an area consists of precentral gyrus and precuneous cortex (marked by green



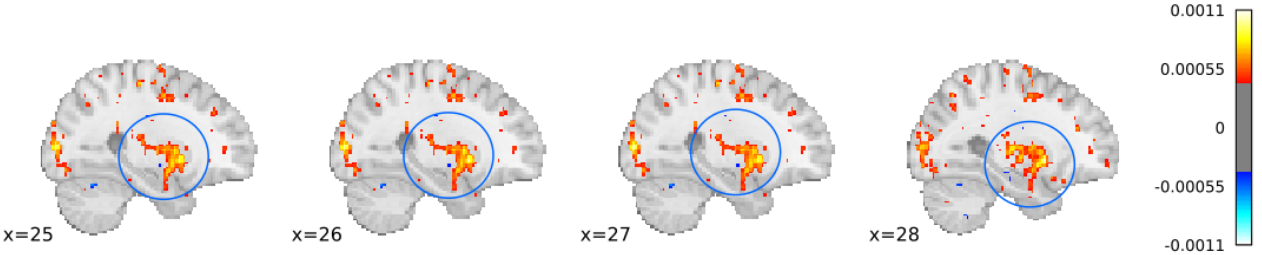
(a) Cerebral Cortex (Left)



(b) Cerebral Cortex (Right)



(c) Putamen (Left)



(d) Putamen (Right)

Figure 5.3: Top 1% model coefficients in magnitude visualised on standard brain atlas. The clusters are considered as identified activation areas and have been marked by circles. The model is trained on the data from the five stop-signal cognitive experiments. The values of the coefficients are shown in the colorbar. Red and blue colours denote positive and negative values, respectively. For each feature, by increasing the value, the final decision will be biassed towards the positive class, i.e. successful stop, if the corresponding coefficient is in red, and the negative class, i.e. unsuccessful stop, if blue. The images are presented in the order from left (less x) to right (greater x). The figures are generated by the Python library Nilearn (Abraham et al., 2014).

circles) are activated. These areas are distributed on both left and right sides, but more active (larger clusters) on the right side. These results are consistent with the meta-analysis across 99 stop-signal tasks on Neurosynth[2] (Yarkoni et al., 2011).

Moreover, some potential new patterns are also discovered. The juxtapositional lobule cortex on the left side of the brain is identified as activation, as shown in Fig. 5.3a (marked by orange circles). However, this area is not a common activation area of stop-signal tasks on Neurosynth. In addition, new activation areas of putamen are also identified, which are nearly equally distributed in both left and right brains, as shown in Fig. 5.3c and 5.3d. However, these areas are only learnt from the five public stop-signal datasets and need to be further investigated by neuroscientists in the future.

5.5 Summary

This chapter proposed a machine learning framework, Covariate-Independence Regularisation (CoIR), for learning generalised patterns across multiple domains. CoIR is motivated by the dependence-based generalisation bound derived in the theoretical analysis in Chapter 4. To reduce the upper bound of generalisation risk, CoIR minimises the empirical prediction risk and domain dependence simultaneously. Based on the CoIR framework, two non-deep algorithms: CoIR_{SVM} and CoIR_{LS} were proposed, and then validated on three different multi-source domain adaptation tasks. On the brain decoding and sentiment analysis tasks, CoIR outperformed all comparing methods. By contrast, deep neural networks obtained better results on the visual object recognition task. The best result on this task is obtained by a new variant of the Moment Matching for Multi-Source Domain Adaptation (Peng et al., 2019), where the domain divergence loss is replaced by CoIR. The experimental results confirmed effectiveness of CoIR (on both non-deep and deep methods), and the "one vs rest" dependence-based generalisation bound.

[2] `https://www.neurosynth.org/analyses/terms/stop%20signal/`

Chapter 6

Covariate-Dependence Learning for Recognising Domain-Specific Patterns

The previous chapters have presented theoretical and experimental results on learning generalised patterns for multi-domain neuroimaging data. This chapter will focus on a different problem: learning specific patterns for a given domain. Specifically, the research question of this chapter is identifying gender-related human brain lateralisation (functionally asymmetry) areas. This question is formulated as a left/right brain hemisphere classification problem, which will be tackled by a novel covariate-dependent learning approach.

6.1 Introduction

Human brains are both structurally and functionally asymmetric, and the lateralisation of brain functions is a popular research topic of neuroscience. Conventionally, human brain lateralisation areas are measured by laterality index (LI) (Seghier, 2008), or identified by comparing the signals of a pair of regions by statistical univariate analysis, e.g. Liégeois et al. (2002). However, the interactions between different functional areas may not be captured. To discover the potential lateralisation patterns, a multivariate machine learning approach will be applied, by constructing brain lateralisation as a left right brain hemisphere classification problem. Then general linear classifiers can be used to learn the patterns of differences between the left and right hemisphere.

Beyond this, a more challenging research question is learning gender-related lateralisation patterns. It is an interesting research problem in neuroscience. Existing studies have shown there are differences in lateralisation between males and females (e.g. (Bechtel & Richardson, 2010)). However, it has not been explored as a classification problem by data-driven approaches. More importantly, male and female are usually nearly equally distributed, which can avoid the potential problem of imbalance. In addition, if a machine learning algorithm can learn gender-specific patterns of lateralisation, then the application can be extended to other interesting research questions, such as handedness, age, or disease-related lateralisation (e.g. Autism disorder).

To tackle this question, a *covariate-dependent machine learning (CoDeML)* framework is proposed and generates an algorithm *covariate-dependent logistic regression (CoDeLR)* by incorporating logistic loss. The effectiveness of CoDeML is validated on the resting-state fMRI data from HCP (Smith et al., 2013). It can successfully learn models with accuracy divergence on data of different genders, compared to a standard logistic regression and general-specific feature separation approaches. By interpreting the learnt model coefficients, some known gender-related lateralisation areas are confirmed, and some potential new patterns are also discovered.

6.2 Methodology

This section will introduce the research problem of gender-related human brain lateralisation and covariate-dependent learning, by building a connection between the neuroscience problem and a mathematical formulation. Statistical theories introduced in Chapter 2 and derived in Chapter 4 will be also applied to analyse the research question of this chapter.

6.2.1 Lateralisation and Brain Hemisphere Classification

To understand (gender-related) brain lateralisation via multivariate machine learning approach is considered in this study, a left/right hemisphere classification problem is constructed. The weights of models that can classify left/right brain hemispheres correctly are hypothesised to represent the importance of each brain area for distinguishing between the

two hemispheres. For this task, machine learning classifiers can be used to train models for classifying brain hemispheres. The discovered patterns can be interpreted by visualising the learnt model weights (or coefficients).

6.2.2 Covariate-Dependent Machine Learning

This subsection presents the covariate-dependent machine learning problem and theoretical analysis, which motivated the *Covariate-Dependent Machine Learning (CoDeML)* framework. Its formulation will be proposed at the end of this subsection.

Problem Formulation

Let $(\mathbf{x}_i, y_i, \mathbf{c}_i)$ be an instance, where $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^d$ denotes an input data vector, $y_i \in \mathcal{Y}$ denotes an output variable (label), $\mathbf{c} \in \mathcal{C} \in \mathbb{R}^v$ denotes an covariate vector, and $i \in [1, m]$, m is the number of all samples. In the context of this work, $\mathbf{x}$ denotes a feature vector that represents one subject's left or right brain hemisphere, y is the label indicates left or right, and $\mathbf{c}$ is a one-hot (zero and one) indicator to represent whether a sample is from a male or female subject (e.g. $c_i = 0$ if male and $c_i = 1$ if female). In this study, a gender-specific model is defined as: has good generalisation performance for target samples specifically, but with high generalisation risk for non-target samples. By this definition, a gender-specific model, e.g. male-specific, should satisfy the two following requirements:

- Making correct predictions on unseen samples of target gender (male), the ideal case is an 100% accuracy.
- Making less correct predictions on unseen samples of non-target gender (female), the ideal case is random level accuracy, e.g. a 50% accuracy for a binary classification problem.

By considering gender as covariate or domain, the learning problem can be called covariate (or domain) dependent machine learning, where the objective can be formulated as: learning a model h_θ with parameters θ specifically for the data of a target covariate. Denoting $\mathrm{R}_t(h_\theta)$ and $\mathrm{R}_{\setminus t}(h_\theta)$as the risk of h_θ on samples from target and non-target covariate, respectively.

For this, we define the objective of a covariate (or domain) specific model h_θ as:

$$\arg\max_{h_\theta} \mathrm{R}_{\setminus t}(h_\theta) - \mathrm{R}_t(h_\theta). \tag{6.1}$$

Hence, the overall objective minimising the generalisation risk for the samples of target covariate, and maximising the generalisation risk for samples of non-target covariate.

Theoretical Analysis

By Theorem 4.6, if target traning samples are labelled, an upper bound of the generalisation risk on non-target samples as following

$$\mathrm{R}_{\setminus t}(h) \leq \hat{\mathrm{R}}_t(h) + \rho_h(\mathbf{X}, \mathbf{CU}) + \Omega, \tag{6.2}$$

where Ω is a constant for fixed data as defined in Theorem 4.6. For unseen target samples, the standard Rademacher generalisation bound applies. By applying Theorem 2.10, the upper bound of the risk on target domain samples is

$$\mathrm{R}_t(h) \leq \hat{\mathrm{R}}_t(h) + \hat{\Re}_{\mathbf{X}^t}(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}}. \tag{6.3}$$

According to the problem of covariate-dependnent learning, accurate predictions for target samples are required. Therefore $\mathrm{R}_t(h)$ needs to be optimised via minimising $\hat{\mathrm{R}}_t(h)$, which is also part of the uppper bound of $\mathrm{R}_{\setminus t}(h)$. Consequently, the upper bound of $\mathrm{R}_{\setminus t}(h)$ can only be increased via the dependence term $\rho_h(\mathbf{X}, \mathbf{CU})$ to achieve the objective of Eq. (6.1) It should be noted that the objective of Eq. (6.1) is not guaranteed to be increased by maximising $\rho_h(\mathbf{X}, \mathbf{CU})$. However, the risk gap is guaranteed to decrease by minimising the dependence on domain covariates. Hence, it is still essential to maximise the statistical dependence for the covariate-dependent learning problem. Based on the above analysis, the covariate-dependent learning can be solved via:

- Maximising the dependence on domain covariates, and
- Minimising the empirical risk on labelled target domain samples.

The theory also indicates that the samples of non-target domain are also important for optimising the dependence, while their label information is not used.

The Framework

On the basis of the learning objective from the theoretical analysis, the Covariate-Dependent Machine Learning (CoDeML) Framework is formulated as following:

$$\arg\min_{h_\theta} \mathcal{L}(h_\theta(\mathbf{X}_\mathrm{t}), \mathbf{y}_\mathrm{t}) + \mu\|h_\theta\|_K^2 - \lambda\rho_h(h_\theta(\mathbf{X}), \mathbf{C}), \tag{6.4}$$

where $\mu \geq 0$ and $\lambda \geq 0$, and $\rho_h(\cdot, \cdot)$ is the simplified HSIC that has been defined in Eq. (5.2). The differences between Eq. (5.1) and Eq. (6.4) are:

- Equation 5.1 minimises the prediction loss on labelled source data, while Equation 6.4 minimises the prediction loss on labelled target data, and
- Equation 5.1 minimises dependence loss, while Equation 6.4 maximises the dependence loss, i.e. the signs before hyper-parameter λ are different.

6.2.3 Covariate-Dependent Logistic Regression (CoDeLR)

Since the statistical dependence needs to be maximised, using maximum likelihood estimation could be a better option to optimise model parameters θ. Therefore, an algorithm for covariate-dependent learning is developed as a new variant of logistic regression, i.e. Covariate-Dependent Logistic Regression (CoDeLR). Denoting $\mathbf{w}$ as a vector of model parameters θ, the probability of target labels given data and model as $\mathbb{P}(\mathbf{y}_\mathrm{t}|\mathbf{w}, \mathbf{X}_\mathrm{t})$, and the probability of dependence given data, model, and covariates as $\mathbb{P}(\mathbf{w})\mathbb{P}(\rho_{sh}|\mathbf{w}, \mathbf{X}, \mathbf{C})$, the overall likelihood to be maximise is

$$\begin{aligned} L(\mathbf{w}) &= \mathbb{P}(\mathbf{y}_\mathrm{t}|\mathbf{w}, \mathbf{X}_\mathrm{t})\mathbb{P}(\mathbf{w})\mathbb{P}(\rho_{sh}|\mathbf{w}, \mathbf{X}, \mathbf{C}) \\ &= \big(\prod_{i=1}^{m_\mathrm{t}} h(\mathbf{x}_i)^{y_i}(1 - h(\mathbf{x}_i))^{(1-y_i)}\big)\frac{1}{\sqrt{2\pi\sigma}} e\big(-\frac{\mathbf{w}^\top\mathbf{w}}{2\sigma^2}\big)\frac{1}{1 + e^{-\mathbf{w}^\top\mathbf{XHLHX}^\top\mathbf{w}}}, \end{aligned} \tag{6.5}$$

where $\mathbf{L} = \mathbf{C}^\top\mathbf{C}$, and $\mathbb{P}(\mathbf{w})$ can be viewed as ℓ_2 regularisation for $\mathbf{w}$ by assuming the parameters following a normal distribution of mean 0 and standard deviation σ. By taking $-\ln$ of Eq. (6.5), and letting μ and λ be the two hyper-parameters for controlling the importance of ℓ_2 norm and covariate dependence regularisation, the objective function becomes

$$\begin{aligned} -\ln L(\mathbf{w}) &= -\sum_{i=1}^{m_t} \big[y_i \ln h(\mathbf{x}_i) + (1-y_i)\ln(1-h(\mathbf{x}_i))\big] + \frac{\mu}{2}\mathbf{w}^\top\mathbf{w} - \frac{\lambda}{2}\ln\sigma(\rho_{sh}(\mathbf{w}^\top\mathbf{X},\mathbf{C})) \\ &= -\sum_{i=1}^{m_t} \big[y_i \ln \frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}}} + (1-y_i)\ln(1-\frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}}})\big] + \frac{\mu}{2}\mathbf{w}^\top\mathbf{w} \\ &\quad - \frac{\lambda}{2}\ln\frac{1}{1+e^{-\mathbf{w}^\top\mathbf{XHLHX}^\top\mathbf{w}}}. \end{aligned} \tag{6.6}$$

Equation 6.6 is convex and can be optimised by gradient descent. To make the optimisation process easier to read, let

$$p_i = h(\mathbf{x}_i) = \frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}_i}}, \tag{6.7}$$

and

$$q = \sigma(\rho_{sh}(\mathbf{w}^\top\mathbf{X},\mathbf{C})) = \frac{1}{1+e^{-\mathbf{w}^\top\mathbf{XHLHX}^\top\mathbf{w}}}, \tag{6.8}$$

and then the gradient of p_i and q can be computed first before deriving the gradient of the whole objective function. Firstly, taking the partial derivatives of p_i w.r.t $\mathbf{w}$ by chain rule, resulting

$$\begin{aligned} \frac{\partial p}{\partial \mathbf{w}_i} &= \big(\frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}_i}}\big)' \\ &= -\big(\frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}_i}}\big)^2(1+e^{-\mathbf{w}^\top\mathbf{x}_i})' \\ &= -\frac{1}{(1+e^{-\mathbf{w}^\top\mathbf{x}_i})^2}e^{-\mathbf{w}^\top\mathbf{x}}(-\mathbf{x}_i) \\ &= \frac{e^{-\mathbf{w}^\top\mathbf{x}_i}}{(1+e^{-\mathbf{w}^\top\mathbf{x}_i})^2}\mathbf{x}_i \\ &= p_i(1-p_i)\mathbf{x}_i. \end{aligned} \tag{6.9}$$

Then similarly, the partial derivatives of q w.r.t $\mathbf{w}$ is

$$\begin{aligned}\frac{\partial q}{\partial \mathbf{w}} &= \big(\frac{1}{1+e^{-\mathbf{w}^\top \mathbf{XHLHX}^\top \mathbf{w}}}\big)' \\ &= -\big(\frac{1}{1+e^{-\mathbf{w}^\top \mathbf{XHLHX}^\top \mathbf{w}}}\big)^2 (1+e^{-\mathbf{w}^\top \mathbf{XHLHX}^\top \mathbf{w}})' \\ &= -\frac{1}{(1+e^{-\mathbf{w}^\top \mathbf{x}})^2} e^{-\mathbf{w}^\top \mathbf{XHLHX}^\top \mathbf{w}} (-2\mathbf{XHLHX}^\top \mathbf{w}) \\ &= \frac{e^{-\mathbf{w}^\top \mathbf{XHLHX}^\top \mathbf{w}}}{(1+e^{-\mathbf{w}^\top \mathbf{x}})^2} (2\mathbf{XHLHX}^\top \mathbf{w}) \\ &= q(1-q)(2\mathbf{XHLHX}^\top \mathbf{w}).\end{aligned} \tag{6.10}$$

Let $J(\mathbf{w}) = -\ln L(\mathbf{w}) = -\sum_{i=1}^{m_\mathrm{t}} \big[y_i \ln p_i + (1-y_i)\ln(1-p_i)\big] + \mu \mathbf{w}^\top \mathbf{w} - \lambda \ln q$, then $J(\mathbf{w})$ become the overall objective and the gradient of $J(\mathbf{w})$ is

$$\begin{aligned}\nabla J(\mathbf{w}) &= -\sum_{i=1}^{m_\mathrm{t}} \big[y_i \frac{p_i(1-p_i)\mathbf{x}_i}{p_i} + (1-y_i)\frac{-p_i(1-p_i)\mathbf{x}_i}{1-p_i}\big] + \mu \mathbf{w} - \lambda \frac{q(1-q)(\mathbf{XHLHX}^\top \mathbf{w})}{q} \\ &= -\sum_{i=1}^{m_\mathrm{t}} (y_i - p_i)\mathbf{x}_i + \mu \mathbf{w} + \lambda(q-1)\mathbf{XHLHX}^\top \mathbf{w} \\ &= (\mathbf{p}_\mathrm{t} - \mathbf{y}_\mathrm{t})\mathbf{X}_\mathrm{t} + \mu \mathbf{w} + \lambda(q-1)\mathbf{XHLHX}^\top \mathbf{w},\end{aligned} \tag{6.11}$$

where $\mathbf{p}_\mathrm{t} = h(\mathbf{X}_\mathrm{t})$. To optimal values of $\mathbf{w}$ can be approximated via updating the following equation in each iteration

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \nabla J(\mathbf{w}^k), \tag{6.12}$$

where k denotes the kth iteration, η is the learning rate (step size). Algorithm 3 is the pseudo-code for CoDeLR.

6.3 Materials and Experiments

This section presents the experiments and results of left/right brain hemisphere classification, which includes: extracting features from resting-state fMRI to present human brain hemisphere, train-test validation strategy for conducting machine learning experiments, experimental results, and an additional study of using domain common and individual feature

Algorithm 3 Covariate Dependent Logistic Regression (CoDeLR)

Input: Input data matrix $\mathbf{X} \in \mathbb{R}^{d \times m}$, label vector $\mathbf{y} \in \mathbb{R}^{m_\mathrm{t}}$, domain covariates, and indices of samples from target covariate (optional, if not given, first m_t samples are assumed to be the labelled target samples).
Hyper-parameters: μ for ℓ_2 norm , λ for HSIC regularisation, and γ for learning rate.
Output: Coefficient vector $\mathbf{w}$.

1: Encode domain covariates into a matrix $\mathbf{C} \in \mathbb{R}^{\hat{d} \times m}$ with one-hot encoding, the construct kernel matrix $\mathbf{L} = \mathbf{C}^\top \mathbf{C}$ and centring matrix $\mathbf{H}$;
2: Add a row of 1s to $\mathbf{X}$
3: Random initialise the parameters of coefficient vector $\mathbf{w} \in \mathbb{R}^{d+1}$;
4: **while** Not converge **do**
5: Compute gradient $\nabla J(\mathbf{w})$ by Eq. (6.11);
6: Update $\mathbf{w}^{k+1} = \mathbf{w}^k - \gamma \nabla J(\mathbf{w}^k)$;
7: **end while**
8: **return** Coefficient vector $\mathbf{w}$.

separation for solving gender-related brain lateralisation.

6.3.1 HCP Resting-state fMRI and Half Brain Connectivity

Resting-state fMRI (rs-fMRI) data from the HCP project (Smith et al., 2013) is used for this brain lateralisation experiment, which consists of four sessions for each subject: one scanned from left to right (LR), one scanned from right to left (RL), and repeated two times in two days. In total, rs-fMRI acquired from 960[1] subjects are selected for experiments.

Figure 6.1 illustrates the data processing workflow for obtaining features to represent the human brain hemisphere from time-series. The time sequences of 246 (123 per half brain) regions of interest were extracted with the BNA parcellations (Fan et al., 2016). Pearson correlation is then computed as the brain region-to-region connectivity, i.e. a 123×123 matrix can be obtained to represent each subject's full brain connectives. Following (Liang et al., 2014), the correlation coefficients are transformed to z scores using Fisher's z transform and then averaged across the scans of RL and LR for each day. To extract half brain features, the columns and rows of the connectivity matrix are reordered by the corresponding ROIs from left to right, and take the within half brain connectivity (the red and blue areas shown in Figure 6.1) as the features used in this study. Table 6.1 summarises the information of

[1]960 is the number of subjects who have the full four sessions of scan available, there are over 1,000 subjects involved in the HCP project in total.

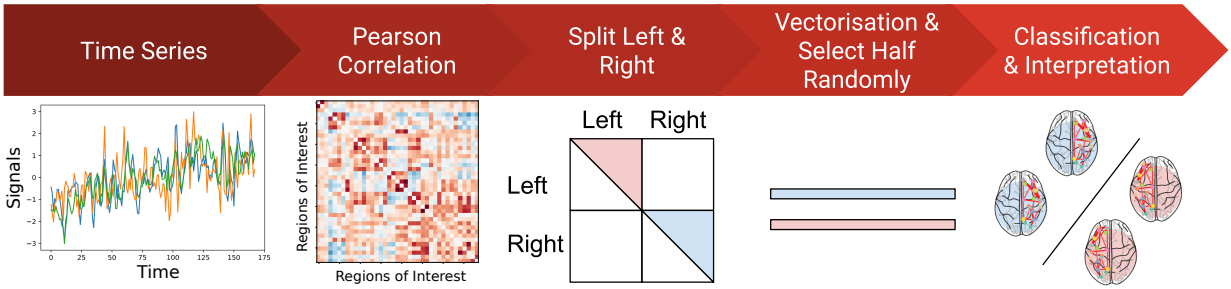


Figure 6.1: Workflow for data processing and machine learning. The resting-state fMRI are processed to obtain within half brain connectivities to represent each subject's left or right brain hemisphere, then machine learning classifiers are trained for classifying the left/right hemisphere. The coefficient of learnt models will be visualised for interpretation.

# Males	# Females	Parcellations Atlas	# ROIs (Full/Half)	# Features
445	515	BNA	246/123	7503

Table 6.1: Summary of processed resting-state function MRI used for experiments, where # denotes "Number of".

used atlas and features.

6.3.2 Experimental Settings

Comparing Baselines

The proposed covariate-dependent logistic regression (CoDeLR) will be compared with a standard logistic regression, i.e. $\lambda = 0$ of Eq. (6.6). Additionally, an unsupervised domain-generic and -specific feature separation approach, **Common Orthogonal Basis Extraction (COBE)** (Zhou et al., 2015), will be used as an alternative baseline. Both of the general and specific features will be used for the brain left/right hemisphere classification problems.

Training-testing Strategy

Since each subject can contribute two half brain hemispheres (one left and one right), the potential correlation between left and right brain of a subject may impact the learning performance of machine learning models. Therefore when using a session for training, two different training-testing strategies are designed to construct a training set:

1. Selecting left half brains from 50% subject randomly, and right brains from the remain-

ing 50% subjects, or

2. Selecting 50% brain hemispheres randomly without considering the impact of subjects, where the number of left and right hemispheres are equal.

Then models trained on the training set will be tested on the remaining hemispheres within the training session and the samples from the held out session. The train-test splits will be repeated 1,000 times for each session and the mean accuracy will be reported later.

Algorithm Setup

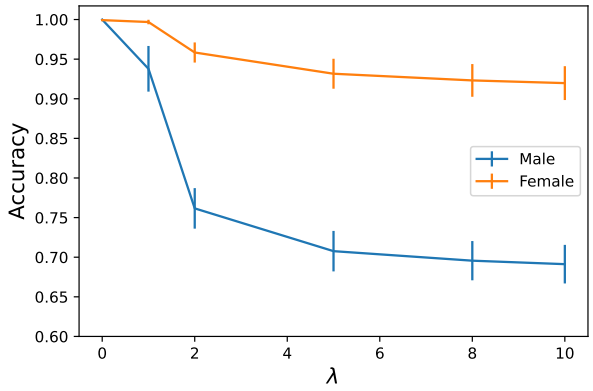
For CoDeLR, only gender is considered as domain covariates in the experiments, and the covariate matrix $\mathbf{C}$ degenerates to a vector in experiment. There are two hyper-parameters of CoDeLR: μ and λ. In this experiment, the value of μ is fixed to be 0.1, and the values of λ in $[0, 1.0, 2.0, 5.0, 8.0, 10.0]$. When $\lambda = 0$, i.e. the dependence on domain covariates does not have any impact on model parameters, CoDeLR will degenerate to a standard logistic regression, which is the baseline for comparison.

For the feature separation approach, domain-generic features will be extracted firstly by COBE. Then to obtain domain-specific features, we follow the strategy introduced in (Lock et al., 2013; Zhou et al., 2015) by subtracting domain-generic features (reconstruction to original space) from cleaned data (as shown in Figure 2.3). As suggested by Zhou et al. (2015), the cleaned data can be obtained by PCA reconstruction. To gain insights of the feature separation approach, the number of common bases for domain-generic features will be varied in range $[2, 3, 5, 10, 15]$, and then observe the changes of classification performance. A linear standard logistic regression will be used as the following classifier for the features extracted by COBE.

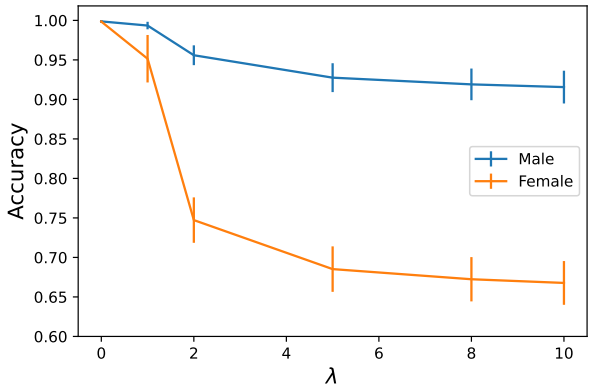
6.3.3 Results

Performance of Covariate-Dependent Machine Learning

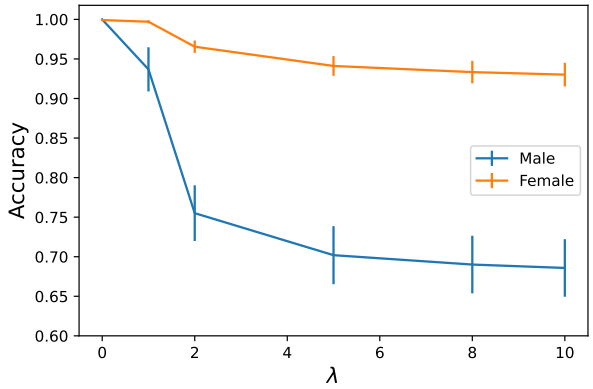
Figure 6.2 presents classification accuracy obtained across different values of λ with the first cross-validation strategy. As shown in all Fig. 6.2a6.2b6.2c6.2d, when $\lambda = 0$, the baseline



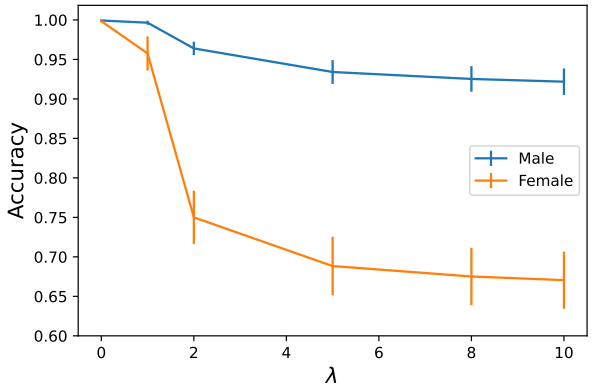
(a) Test accuracy on the held out samples from the session for training. Models are trained on labelled female and unlabelled male samples.



(b) Test accuracy on the held out samples from the session for training. Models are trained on labelled male and unlabelled female samples.



(c) Test accuracy on the samples from the held out session. Models are trained on labelled female and unlabelled male samples.

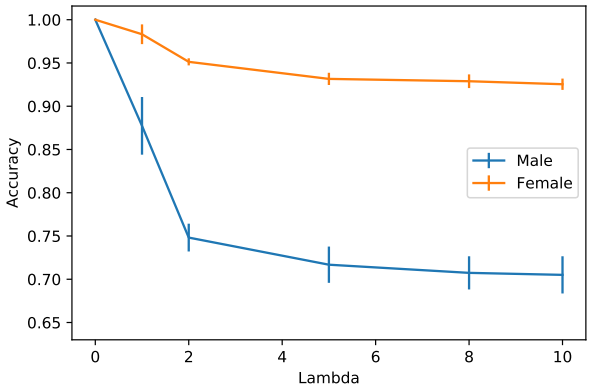


(d) Test accuracy on the samples from the held out session. Models are trained on labelled male and unlabelled female samples.

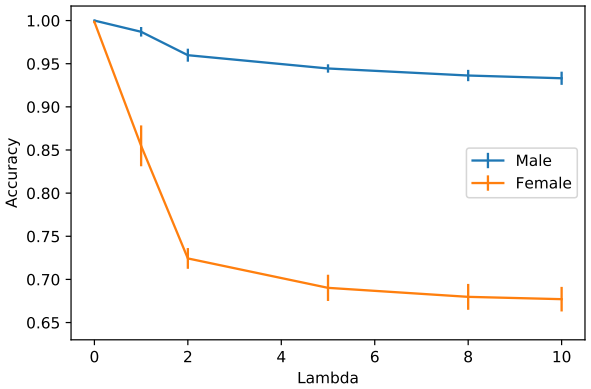
Figure 6.2: Averaged left/right brain hemisphere classification accuracy with standard deviations. When one session is used for training, each subject contributes either left or right hemisphere to training set. The learnt models are tested on the samples remaining within and session, and of the held out session.

logistic regression obtained near 100% classification accuracy on both male and female test samples, even the models are trained on male (or female) samples only. By increasing the value of λ, i.e. the importance of dependence on the covariate of gender, a gap between accuracy on male and female test samples is created. When $\lambda > 2$, a 20% accuracy difference can be observed for both male-specific and female-specific learning, which shows the proposed CoDeLR can help to learn more specific models.

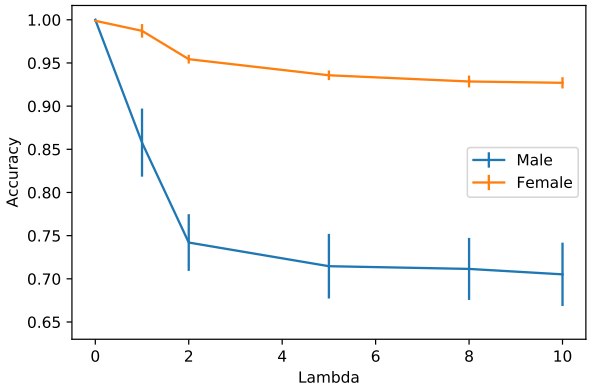
Figure 6.3 reports the classification accuracy with the second cross-validation strategy. In general, the results are very similar to Fig. 6.2. By performing significance test between the



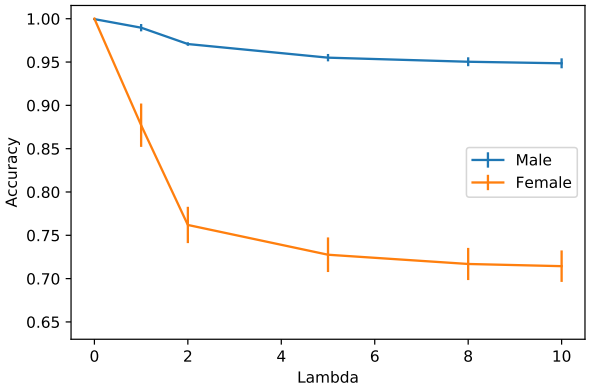
(a) Test accuracy on the held out samples from the session for training. Models are trained on labelled female and unlabelled male samples.



(b) Test accuracy on the held out samples from the session for training. Models are trained on labelled male and unlabelled female samples.



(c) Test accuracy on the samples from the held out session. Models are trained on labelled female and unlabelled male samples.

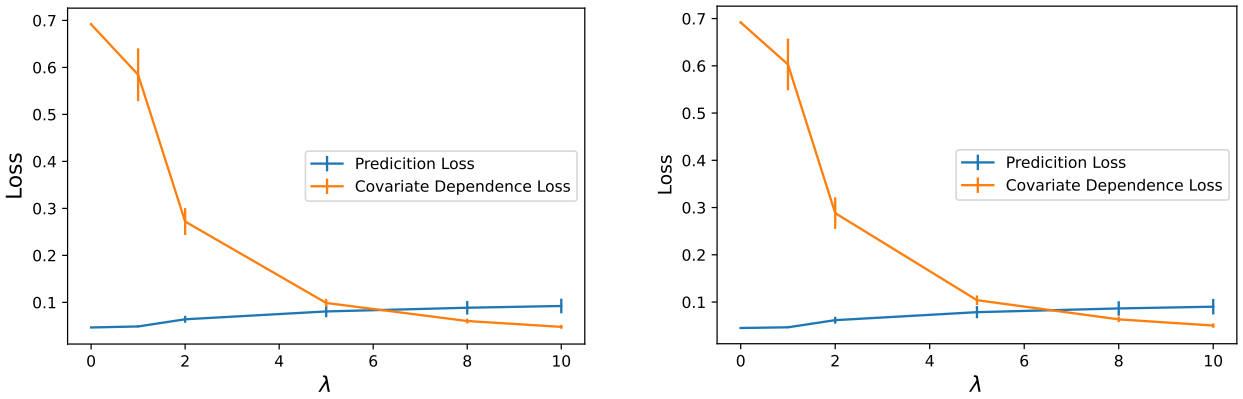


(d) Test accuracy on the samples from the held out session. Models are trained on labelled male and unlabelled female samples.

Figure 6.3: Averaged left/right brain classification accuracy with standard deviations. Models are trained on the random selected 50% of brain hemispheres from one session (number of left and half right hemispheres are equal). The learnt model will be tested on the remaining 50% of the samples within and session, and the held out session.

accuracy obtained by the two cross-validation strategies, the p-value is less than 0.05, which means statistically there is no significant difference between the two strategies, i.e. the paired brain hemispheres do not have impact on machine learning models in terms of accuracy.

Figure 6.4 presents the average training loss across different values of the hyper-parameter λ, for both male- and female-specific models. It is obvious that with the increase of λ, the loss of covariate dependence drop significantly, especially when $\lambda \leq 5$. While the larger value of λ can also lead to a mild growth in prediction loss. By comparing the results in Fig. 6.4a and Fig. 6.4b, there is no significant difference. To sum up, all the results in Fig. 6.4 are



(a) Loss of female-specific models.

(b) Loss of male-specific models.

Figure 6.4: Averaged training losses over the hyper-parameter λ.

consistent with the findings in Fig. 6.2 and 6.3.

Performance of General and Specific Feature Separation

Figure 6.5 reports the brain hemisphere classification results across number of common basis for domain-generic features. For both male- and female-specific features, highest accuracy are around 95%, which are obtained when the number of common basis equals 2. Then it dropped to below 70% when common basis increased to 3, and kept decreasing with the growing of common basis.

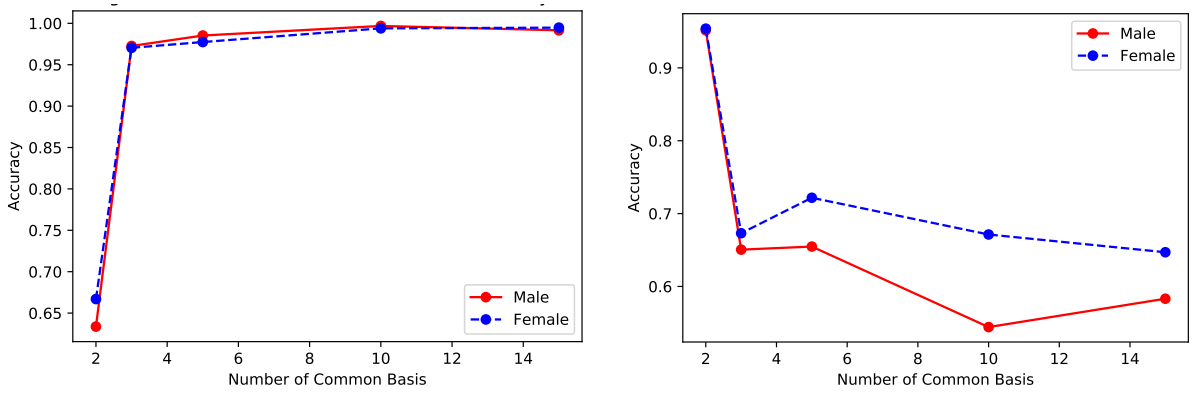
On the contrary, an opposite trend is shown on the classification performance with general features across genders. When the number of common basis equals 2, the accuracy is the lowest at 65%. When common basis ≥ 3, the accuracy kept steadily above 95 %.

6.4 Discussions

6.4.1 Further Analysis of Experimental Results

Covariate-Dependent Learning: from Theory to Experimental Results

The effectiveness of regularising covariate dependence has been validated in the previous section. With the increasing of the hyper-parameter λ, i.e. larger impact of covariate-dependence regularisation, the accuracy gap between data of different genders become larger.



(a) Accuracy obtained with the domain-generic features extracted by COBE.

(b) Accuracy obtained with the domain-specific features extracted by COBE.

Figure 6.5: Brain hemisphere classification accuracy obtained on general and specific features extracted by *Common Orthogonal Basis Extraction (COBE)* Zhou et al. (2015) across different numbers of common basis for domain-generic features.

This is consistent with the theoretical analysis in Section 6.2, which indicates higher covariate-dependence can lead to larger upper bounds of the accuracy gap.

It should be emphasised here that it is the upper bound of the gap increased with the growth of covariate dependence but not the real gap. Another interesting observation from Figure 6.2 and 6.3 can support the theory. The standard deviations for non-target gender (the lines below) are larger than the ones of target gender (the lines above). This can be evidence to support the previous theoretical analysis. In the objective functions (Eq. (6.5) and (6.6)), the prediction loss on target domain is optimised, and therefore the risk on non-target domain can be unstable and with a higher upper bound, which is reflected as the higher standard deviation.

Feature Separations

Figure 6.5 shows the informative features for classifying left/right brains that cannot be preserved for both domain-generic and -specific features. The results obtained by CoDeML have shown there exist gender-generic and -specific patterns of brain lateralisation, which are considered as discriminative information for brain hemisphere classification. Therefore, the results obtained by COBE suggest that the general and specific lateralisation patterns were not successfully separated. This could be another evidence to prove that using label

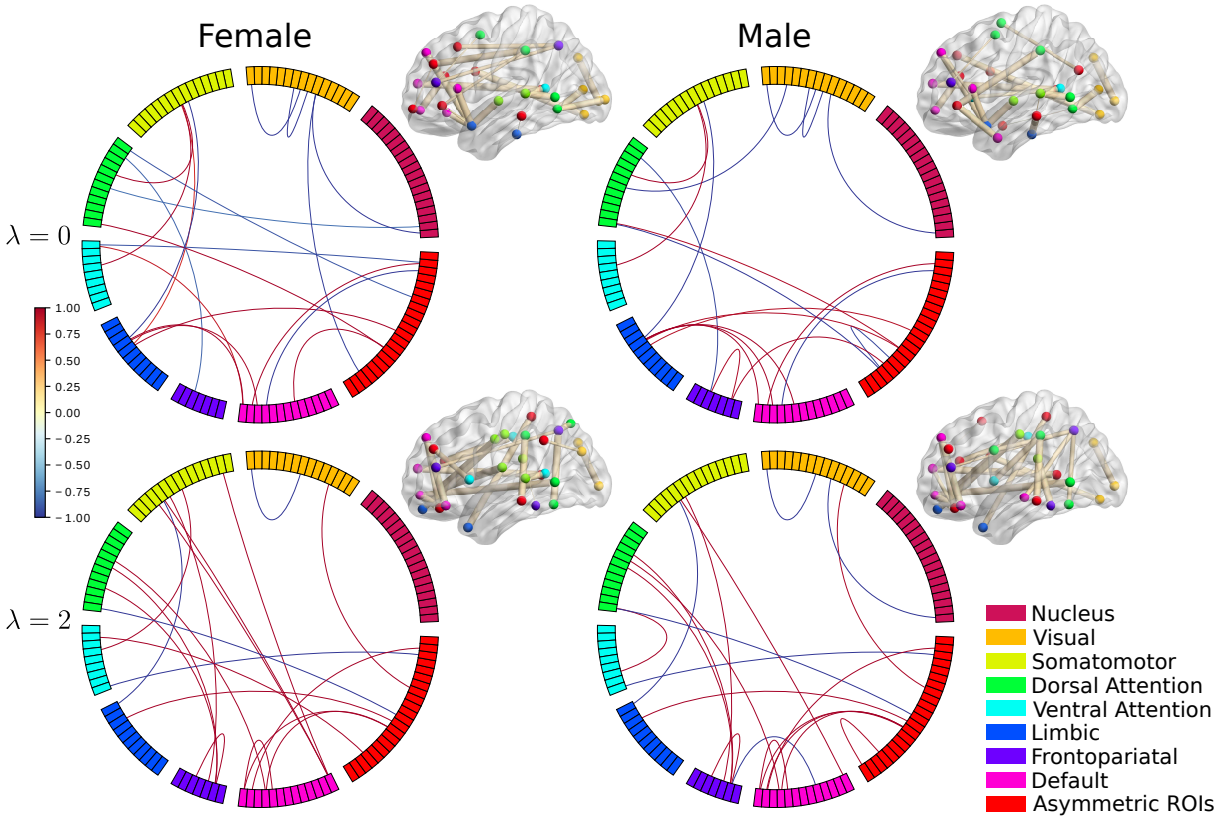


Figure 6.6: Top 20 discriminative features for left/right brain hemisphere classification, where the rank is determined by the frequency of the correspoding weight coefficients are in top 50 by magnitude over the models obtained from the 1,000 random train-test splits. Each circle represents a brain hemisphere, where each small block represents a ROI and each chord represents the connection between two regions, which are also presented in the top right figures for locations in the brain. The edge of each circle is divided into nine arches that represent the seven subnetworks defined in the Yeo7 brain functional network parcellation (Yeo et al., 2011) plus nucleus and "Asymmetric ROIs", which is the homotopic (same position in the two hemispheres) ROIs but in different functional subnetwork. Each subnetwork is denoted by a colour. The colour of chords denotes the frequency and the sign of their correponding weights (Red/1.0: top positive weights and frequency=1000; Blue/-1.0: top negative weights and frequency=1000). For more figures and details, please see Appendix B.

information can improve the effectiveness of domain-aware learning.

6.4.2 Model Visualisation and Interpretation

Figure. 6.6 demonstrates the frequency of top 20 discriminative features (by magnitude of corresponding weight coefficients) over the models obtained in the 1,000 random train-test

splits. From this figure, we can observe:

- When $\lambda = 0$, the patterns learnt from labelled male and female samples are different, despite the generalision error on the non-target gender samples is low in experiments.
- When $\lambda = 2$, the "male-specific" and "female-specific" patterns learnt are still visually similar, althought there are accuracy gaps reported in the previouse section. It should be noted here that although some of the connections between subnetworks are similar, the exact nodes/ROIs of the connections are different, e.g. the connections between default mode network (DMN) and Somatomotor, and the connections between front-parietal network (FPN) and dorsal attention network (DAN).
- For patterns learnt with different values of λ but the same target gender, the similarity is relatively low. This difference can only be observed when $\lambda \in [0, 2]$. When $\lambda \geq 2$, the patterns become stable and similar. Please see Appendix B for more details.
- The inter-subnetwork connections is much more than intra-subnetwork connections.

These observation can be a guidence for future neuroscience studies to understand gender-related brain lateralisation. Especially the areas of DAN, DMN, somatomotor, and asymmetric ROIs, which are highly weighted and need to be further explored.

6.5 Summary

This chapter explored a research question which is opposite to the previous chapters: learning domain-specific patterns. For this purpose, a novel framework *Covariate Dependent Machine Learning (CoDeML)* is proposed, which is also based on the theoretical study in Chapter 4. A new variant of logistic regression, *Covariate-Dependent Logistic Regression (CoDeLR)* is developed under the framework, by minimising the empirical risk on target domain, and maximising the dependence on domain covariates. There are two main differences from the framework in the previous chapter: 1) the empirical prediction risk is minimised on target samples instead of source (non-target) samples; 2) the dependence on domain covariates needs to be maximised but not minimised. Experimental results on HCP resting-state fMRI data

show CoDeLR can successfully learn gender-specific models on brain hemisphere classification tasks. A classification accuracy is created by increasing the importance of the covariate dependence regularisation. By interpreting the learnt model coefficients, meaningful lateralisation patterns and insights are discovered.

Chapter 7

Conclusion and Future Work

In this thesis, three domain-aware learning approaches have been presented for learning general or specific patterns from multi-domain neuroimaging data, combined with a new derived dependence-based generalisation bound. Two empirical studies of learning general patterns on brain decoding tasks and one research on learning gender-specific patterns of brain lateralisation confirmed the effectiveness of domain-aware learning approaches. A summary of the thesis and potential future works will be presented in the following.

7.1 Summary of Thesis

Standard machine learning approaches face great challenges of high dimensionality, low signal-to-noise ratio, and small sample size in functional neuroimage analysis. The emergence of public neuroimaging datasets makes analysing large-scale data from multiple domains possible. This thesis explored using domain-aware learning approaches to tackle the above challenges and answered two research questions: 1) How to learn general patterns across domains, and 2) How to learn domain-specific patterns.

Chapter 3 presented our first attempt, which is a two-stage framework with feature and classifier adaptation developed under the semi-supervised domain adaptation setting. Experimental results confirmed the effectiveness of this two-stage adaptation approach on brain decoding tasks constructed on OpenNeuro data.

Based on the empirical study above, in Chapter 4 we derived a dependence-based gen-

eralisation bound to guide the design of domain-aware learning algorithms for the more challenging unsupervised multi-source domain adaptation problem. This theoretical result leads to a covariate-independence regularisation (CoIR) framework in Chapter 5 by viewing domain information as covariates. Incorporating hinge and least squares loss in this framework generates two algorithms: covariate-dependence regularised support vector machine and least-squares classifier. We studied both algorithms on not only brain decoding but also textual sentiment classification and visual object recognition to investigate their efficacy across multiple applications. Experimental results validated the superiority of CoIR over competing methods.

Motivated by both the classic and dependence-based generalisation bounds, in Chapter 6 we propose a covariate-dependent machine learning framework to learn domain-specific patterns. Under this framework, we employ logistic loss to obtain covariate-dependent logistic regression, which successfully learnt gender-specific patterns of brain lateralisation on brain hemisphere classification tasks with data from Human Connectome Project.

All domain-aware learning approaches proposed in this thesis are primarily designed to produce linear, interpretable models for neuroimage classification tasks. By visualising the linear model coefficients in feature space, we validated existing findings of neuroscience and identified new, meaningful patterns to offer additional insights.

As shown theoretically and empirically in this thesis, interpretable domain-aware learning offers feasible ways to learn interpretable general or specific patterns from multi-domain neuroimaging data for neuroscientists to gain insights.

7.2 Future Works

7.2.1 Further Development of Domain-Aware Learning

Covariate-Independence Regularisation for Deep Neural Networks

In Chapter 5, we developed a new variant of deep multi-source domain adaptation network, by incorporating the proposed covariate-independence regularisation (CoIR) to an existing network architecture Moment Matching for Multi-Source Domain Adaptation (M^3SDA) (Peng

et al., 2019). The developed variant obtained the best result on the visual object recognition tasks, which is even better than the results with a deeper backbone network reported in the original paper. This shows the potential of CoIR application for deep neural networks, which can be an interesting research question to explore.

Further Theoretical Studies

In this thesis, Chapter 2 presented several generalisation bounds, and Chapter 4 derived a new generalisation bound. Based on the previous theoretical analysis, "source-combine" $\leq$ "one vs rest" $\leq$ "one vs one" in terms of tightness. However, according to the experimental results on multi-source domain adaptation in this thesis (and other studies), the source-combine, or single-source methods are usually performed the worst among domain adaptation methods, and "one vs rest" methods outperformed "one vs one" methods on average, i.e. the tighter bounds does not guarantee better performance. Therefore, further theoretical analysis is needed to explain this phenomenon and guide the applications in the future.

7.2.2 New Applications of Domain-Aware Learning Algorithms

Cardiac MRI Analysis

We have had some successful experience (Swift et al., 2021; Alabed et al., 2022) of applying machine learning methods to Cardiac MRI (CMRI) data collected locally in Sheffield, with good classification results and high interpretability. In the future, there will be more data collected from our collaborating hospitals across the world. Domain-aware learning can be used for such data to tackle the potential heterogeneity by viewing different hospitals or other covariates, such as race, as domains.

Computer Vision or Natural Language Processing Tasks

Deep neural networks are powerful on computer vision and natural language processing tasks, which requires lower interpretability. Therefore, the research of covariate-independence regularised networks introduced in Chapter 5 has shown some promising results. In the future, by combining the advantage of domain-aware learning and feature representation learning of

deep neural networks (convolutional neural network, transformer (Vaswani et al., 2017), more studies can be conducted on these tasks, and therefore the potential impact of domain-aware learning approaches in general machine learning applications can be improved.

Other Domain-Specific Brain Lateralisation Studies

As mentioned in Chapter 6, applications of covariate-dependent can be extended to other interesting research questions in neuroscience, such as handedness, age, or disease-related lateralisation. These researches can bring neuroscientists deeper insights about human brain functions or brain disorders, with the potential serving for better healthcare.

Bibliography

Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., & Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, *8*, 14.

Alabed, S., Uthoff, J., Zhou, S., Garg, P., Dwivedi, K., Alandejani, F., Gosling, R., Schobs, L., Brook, M., Shahin, Y., Capener, D., Johns, C. S., Wild, J. M., Rothman, A. M. K., van der Geest, R. J., Condliffe, R., Kiely, D. G., Lu, H., & Swift, A. J. (2022). Machine learning cardiac-MRI features predict mortality in newly diagnosed pulmonary arterial hypertension. *European Heart Journal - Digital Health*. Ztac022.
URL `https://doi.org/10.1093/ehjdh/ztac022`

Allen, E. A., Damaraju, E., Plis, S. M., Erhardt, E. B., Eichele, T., & Calhoun, V. D. (2014). Tracking whole-brain connectivity dynamics in the resting state. *Cerebral Cortex*, *24*(3), 663–676.

Aron, A. R., Behrens, T. E., Smith, S., Frank, M. J., & Poldrack, R. A. (2007). Triangulating a cognitive control network using diffusion-weighted magnetic resonance imaging (MRI) and functional MRI. *Journal of Neuroscience*, *27*(14), 3743–3752.

Aydore, S., Thirion, B., & Varoquaux, G. (2019). Feature grouping as a stochastic regularizer for high-dimensional structured data. In *ICML*, (pp. 385–394).

Bartlett, P. L., Boucheron, S., & Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, *48*(1), 85–113.

Bechtel, W., & Richardson, R. C. (2010). *Discovering complexity: Decomposition and localization as strategies in scientific research*. MIT press.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, *7*(Nov), 2399–2434.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1-2), 151–175.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In *Neurips*, (pp. 137–144).

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, (pp. 17–36). JMLR Workshop and Conference Proceedings.

Berlinet, A., & Thomas-Agnan, C. (2011). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer Science & Business Media.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (pp. 440–447).

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., & Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, *22*(14), e49–e57.

Brett, M., Hanke, M., Côté, M.-A., Markiewicz, C., Ghosh, S., Wassermann, D., Gerhard, S., Larson, E., Lee, G. R., Halchenko, Y., Kastman, E., M, C., Morency, F. C., moloney, Rokem, A., Cottaar, M., Millman, J., jaeilepp, Gramfort, A., Vincent, R. D., McCarthy, P., van den Bosch, J. J., Subramaniam, K., Nichols, N., embaker, markhymers, chaselgrove, Basile, Oosterhof, N. N., & Nimmo-Smith, I. (2017). nipy/nibabel: 2.2.0.
URL `https://doi.org/10.5281/zenodo.1011207`

Brodersen, K. H., Schofield, T. M., Leff, A. P., Ong, C. S., Lomakina, E. I., Buhmann, J. M., & Stephan, K. E. (2011). Generative embedding for model-based classification of fMRI data. *PLoS Comput Biol*, *7*(6), e1002079.

Cao, Y., Long, M., & Wang, J. (2018). Unsupervised domain adaptation with distribution matching machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, (pp. 2795–2802).

Chen, M., Xu, Z., Weinberger, K. Q., & Sha, F. (2012). Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, (pp. 1627–1634).

Chen, P.-H. C., Chen, J., Yeshurun, Y., Hasson, U., Haxby, J., & Ramadge, P. J. (2015). A reduced-dimension fMRI shared response model. In *NeurIPS*, (pp. 460–468).

Cheng, B., Liu, M., Shen, D., Li, Z., Zhang, D., Initiative, A. D. N., et al. (2017). Multi-domain transfer learning for early diagnosis of alzheimer's disease. *Neuroinformatics*, (pp. 1–18).

Cheng, B., Liu, M., Suk, H.-I., Shen, D., Zhang, D., Initiative, A. D. N., et al. (2015a). Multimodal manifold-regularized transfer learning for mci conversion prediction. *Brain imaging and behavior*, *9*(4), 913–926.

Cheng, B., Liu, M., Zhang, D., Munsell, B. C., & Shen, D. (2015b). Domain transfer learning for mci conversion prediction. *IEEE Transactions on Biomedical Engineering*, *62*(7), 1805–1817.

Cheng, B., Zhang, D., & Shen, D. (2012). Domain transfer learning for mci conversion prediction. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, (pp. 82–90).

Cheng, W., Palaniyappan, L., Li, M., Kendrick, K. M., Zhang, J., Luo, Q., Liu, Z., Yu, R., Deng, W., Wang, Q., et al. (2015c). Voxel-based, brain-wide association study of aberrant functional connectivity in schizophrenia implicates thalamocortical circuitry. *NPJ Schizophrenia*, *1*, 15016.

Chu, W.-S., De la Torre, F., & Cohn, J. F. (2017). Selective transfer machine for personalized facial expression analysis. *TPAMI*, *39*(3), 529–545.

Cox, C. R., & Rogers, T. T. (2021). Finding distributed needles in neural haystacks. *Journal of Neuroscience*, *41*(5), 1019–1032.

Cox, D. D., & Savoy, R. L. (2003). Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, *19*(2), 261–270.

Craddock, C., Benhajali, Y., Chu, C., Chouinard, F., Evans, A., Jakab, A., Khundrakpam, B. S., Lewis, J. D., Li, Q., Milham, M., et al. (2013). The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Neuroinformatics*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, (pp. 248–255). Ieee.

Dou, Q., de Castro, D. C., Kamnitsas, K., & Glocker, B. (2019). Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, (pp. 6447–6458).

Fan, L., Li, H., Zhuo, J., Zhang, Y., Wang, J., Chen, L., Yang, Z., Chu, C., Xie, S., Laird, A. R., et al. (2016). The human brainnetome atlas: a new brain atlas based on connectional architecture. *Cerebral Cortex*, *26*(8), 3508–3526.

Finn, E. S., Shen, X., Scheinost, D., Rosenberg, M. D., Huang, J., Chun, M. M., Papademetris, X., & Constable, R. T. (2015). Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nature Neuroscience*, *18*(11), 1664–1671.

Fox, M. D., & Raichle, M. E. (2007). Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature Reviews Neuroscience*, *8*(9), 700–711.

Friston, K. J., Fletcher, P., Josephs, O., Holmes, A., Rugg, M., & Turner, R. (1998). Event-related fMRI: characterizing differential responses. *NeuroImage*, *7*(1), 30–40.

Friston, K. J., Holmes, A. P., Poline, J., Grasby, P., Williams, S., Frackowiak, R. S., & Turner, R. (1995). Analysis of fmri time-series revisited. *Neuroimage*, *2*(1), 45–53.

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., & Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Human Brain Mapping*, *2*(4), 189–210.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, *17*, 59:1–59:35.

Ghafoorian, M., Mehrtash, A., Kapur, T., Karssemeijer, N., Marchiori, E., Pesteie, M., Guttmann, C. R., de Leeuw, F.-E., Tempany, C. M., van Ginneken, B., et al. (2017). Transfer learning for domain adaptation in MRI: Application in brain lesion segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017*, (pp. 516–524). Springer.

Gheiratmand, M., Rish, I., Cecchi, G. A., Brown, M. R., Greiner, R., Polosecki, P. I., Bashivan, P., Greenshaw, A. J., Ramasubbu, R., & Dursun, S. M. (2017). Learning stable and predictive network-based patterns of schizophrenia and its clinical symptoms. *NPJ schizophrenia*, *3*(1), 22.

Gong, B., Shi, Y., Sha, F., & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, (pp. 2066–2073). IEEE.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., & Schölkopf, B. (2016). Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, (pp. 2839–2848).

Gonzalez-Castillo, J., Hoy, C. W., Handwerker, D. A., Robinson, M. E., Buchanan, L. C., Saad, Z. S., & Bandettini, P. A. (2015). Tracking ongoing cognition in individuals using brief, whole-brain functional connectivity patterns. *Proceedings of the National Academy of Sciences*, *112*(28), 8762–8767.

Gorgolewski, K., Esteban, O., Schaefer, G., Wandell, B., & Poldrack, R. (2017). Openneuro - a free online platform for sharing and analysis of neuroimaging data. In *OHBM*, (p. 1677).

Gorgolewski, K. J., Varoquaux, G., Rivera, G., Schwarz, Y., Ghosh, S. S., Maumet, C., Sochat, V. V., Nichols, T. E., Poldrack, R. A., Poline, J.-B., et al. (2015). Neurovault. org: a web-based repository for collecting and sharing unthresholded statistical maps of the human brain. *Frontiers in neuroinformatics*, *9*, 8.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *ALT*, (pp. 63–77). Springer.

Guo, J., Shah, D., & Barzilay, R. (2018). Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (pp. 4694–4703).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, (pp. 770–778).

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*(301), 13–30.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, *13*(4), 411–430.

Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., & Smith, S. M. (2012). FSL. *NeuroImage*, *62*(2), 782–790.

Jiang, W., Zavesky, E., Chang, S.-F., & Loui, A. (2008). Cross-domain learning methods for high-level visual concept classification. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, (pp. 161–164). IEEE.

Jitkrittum, W., Szabó, Z., & Gretton, A. (2017). An adaptive test of independence with analytic kernel embeddings. In *International Conference on Machine Learning*, (pp. 1742–1751). PMLR.

Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, *47*(5), 1902–1914.

Koltchinskii, V., & Panchenko, D. (2000). Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, (pp. 443–457). Springer.

Kriegeskorte, N., Goebel, R., & Bandettini, P. (2006). Information-based functional brain mapping. *Proceedings of the National Academy of Sciences*, *103*(10), 3863–3868. URL `http://dx.doi.org/10.1073/pnas.0600244103`

Kunda, M., Zhou, S., Gong, G., & Lu, H. (2020). Improving multi-site autism classification based on site-dependence minimisation and second-order functional connectivity. *bioRxiv*.

LaConte, S., Strother, S., Cherkassky, V., Anderson, J., & Hu, X. (2005). Support vector machines for temporal classification of block design fmri data. *NeuroImage*, *26*(2), 317–329.

Li, H., Parikh, N. A., & He, L. (2018). A novel transfer learning approach to enhance deep neural network classification of brain functional connectomes. *Frontiers in Neuroscience*, *12*, 491.

Liang, Z., King, J., & Zhang, N. (2014). Neuroplasticity to a single-episode traumatic stress revealed by resting-state fmri in awake rats. *Neuroimage*, *103*, 485–491.

Liégeois, F., Connelly, A., Salmond, C., Gadian, D. G., Vargha-Khadem, F., & Baldeweg, T. (2002). A direct test for lateralization of language activation using fmri: comparison with invasive assessments in children with epilepsy. *NeuroImage*, *17*(4), 1861–1867.

Lock, E. F., Hoadley, K. A., Marron, J., & Nobel, A. B. (2013). Joint and individual variation explained (jive) for integrated analysis of multiple data types. *Annals of Applied Statistics*, *7*(1), 523–542.

Long, M., Cao, Y., Cao, Z., Wang, J., & Jordan, M. I. (2019). Transferable representation learning with deep adaptation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*, 3071–3085.

Long, M., Wang, J., Ding, G., Pan, S. J., & Philip, S. Y. (2013a). Adaptation regularization: A general framework for transfer learning. *TKDE*, *26*(5), 1076–1089.

Long, M., Wang, J., Ding, G., Sun, J., & Yu, P. S. (2013b). Transfer feature learning with joint distribution adaptation. In *ICCV*, (pp. 2200–2207).

Lu, H., Liu, X., Turner, R., Bai, P., Koot, R. E., Zhou, S., Chasmai, M., & Schobs, L. (2021). Pykale: Knowledge-aware machine learning from multiple sources in python. *arXiv preprint arXiv:2106.09756*.

Mandelkow, H., De Zwart, J. A., & Duyn, J. H. (2016). Linear discriminant analysis achieves high classification accuracy for the bold fmri response to naturalistic movie stimuli. *Frontiers in human neuroscience*, *10*, 128.

Marden, J. I. (2014). *Analyzing and Modeling Rank Data*. Chapman and Hall/CRC.

Mensch, A., Mairal, J., Bzdok, D., Thirion, B., & Varoquaux, G. (2017). Learning neural representations of human cognition across many fMRI studies. In *NeurIPS*, (pp. 5883–5893).

Ng, B., Vahdat, A., Hamarneh, G., & Abugharbieh, R. (2010). Generalized sparse classifiers for decoding cognitive states in fmri. In *International Workshop on Machine Learning in Medical Imaging*, (pp. 108–115). Springer.

Norman, K. A., Polyn, S. M., Detre, G. J., & Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in cognitive sciences*, *10*(9), 424–430.

Ogawa, S., Lee, T.-M., Kay, A. R., & Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *PNAS*, *87*(24), 9868–9872.

Pakravan, M., & Shamsollahi, M. B. (2018). Extraction and automatic grouping of joint and individual sources in multisubject fmri data using higher order cumulants. *IEEE journal of biomedical and health informatics*, *23*(2), 744–757.

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, *22*(2), 199–210.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *TKDE*, *22*(10), 1345–1359.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (pp. 1406–1415).

Poldrack, R., Kittur, A., Kalar, D., Miller, E., Seppa, C., Gil, Y., Parker, D., Sabb, F., & Bilder, R. (2011). The cognitive atlas: Toward a knowledge foundation for cognitive neuroscience. *Frontiers in Neuroinformatics*, *5*, 17.

Poldrack, R. A. (2007). Region of interest analysis for fMRI. *Social Cognitive and Affective Neuroscience*, *2*(1), 67–70.

Poldrack, R. A., Barch, D. M., Mitchell, J., Wager, T., Wagner, A. D., Devlin, J. T., Cumba, C., Koyejo, O., & Milham, M. (2013). Toward open sharing of task-based fMRI data: the OpenfMRI project. *Frontiers in Neuroinformatics*, *7*, 12.

Rao, N., Cox, C., Nowak, R., & Rogers, T. T. (2013). Sparse overlapping sets lasso for multitask learning and its application to fMRI analysis. In *NeurIPS*, (pp. 2202–2210).

Redko, I., Morvant, E., Habrard, A., Sebban, M., & Bennani, Y. (2020). A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*.

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *European Conference on Computer Vision*, (pp. 213–226). Springer.

Schirmer, M. D., Venkataraman, A., Rekik, I., Kim, M., Mostofsky, S. H., Nebel, M. B., Rosch, K., Seymour, K., Crocetti, D., Irzan, H., et al. (2021). Neuropsychiatric disease classification using functional connectomics-results of the connectomics in neuroimaging transfer learning challenge. *Medical image analysis*, *70*, 101972.

Schölkopf, B., Herbrich, R., & Smola, A. J. (2001). A generalized representer theorem. In *COLT*, (pp. 416–426). Springer.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, *10*(5), 1299–1319.

Seghier, M. L. (2008). Laterality index in functional mri: methodological issues. *Magnetic Resonance Imaging*, *26*(5), 594–601.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Smith, S. M., Beckmann, C. F., Andersson, J., Auerbach, E. J., Bijsterbosch, J., Douaud, G., Duff, E., Feinberg, D. A., Griffanti, L., Harms, M. P., et al. (2013). Resting-state fMRI in the human connectome project. *NeuroImage*, *80*, 144–168.

Smith, S. M., Miller, K. L., Salimi-Khorshidi, G., Webster, M., Beckmann, C. F., Nichols, T. E., Ramsey, J. D., & Woolrich, M. W. (2011). Network modelling methods for fMRI. *NeuroImage*, *54*(2), 875–891.

Song, L. (2008). *Learning via Hilbert space embedding of distributions Ph. D*. Ph.D. thesis, thesis, School of Information Technologies, Univ. Sydney.

Swift, A. J., Lu, H., Uthoff, J., Garg, P., Cogliano, M., Taylor, J., Metherall, P., Zhou, S., Johns, C. S., Alabed, S., et al. (2021). A machine learning cardiac magnetic resonance approach to extract disease features and automate pulmonary arterial hypertension diagnosis. *European Heart Journal-Cardiovascular Imaging*, *22*(2), 236–245.

Tahmoresnezhad, J., & Hashemi, S. (2017). Visual domain adaptation via transfer feature learning. *Knowledge and Information Systems*, *50*(2), 585–605.

Toiviainen, P., Alluri, V., Brattico, E., Wallentin, M., & Vuust, P. (2014). Capturing the musical brain with lasso: Dynamic decoding of musical features from fmri data. *Neuroimage*, *88*, 170–180.

Tong, F., & Pratte, M. S. (2012). Decoding patterns of human brain activity. *Annual Review of Psychology*, *63*, 483–509.

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

Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S. W., et al. (2012). The human connectome project: a data acquisition perspective. *NeuroImage*, *62*(4), 2222–2231.

Vapnik, V. (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media.

Vapnik, V., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, *16*(2), 264–280.

Varoquaux, G., Baronnet, F., Kleinschmidt, A., Fillard, P., & Thirion, B. (2010a). Detection of brain functional-connectivity difference in post-stroke patients using group-level covariance modeling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (pp. 200–208). Springer.

Varoquaux, G., Gramfort, A., baptiste Poline, J., & Thirion, B. (2010b). Brain covariance selection: better individual functional connectivity models using population prior. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.) *Advances in Neural Information Processing Systems 23*, (pp. 2334–2342). Curran Associates, Inc.

Varoquaux, G., Gramfort, A., Pedregosa, F., Michel, V., & Thirion, B. (2011). Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Biennial*

International Conference on Information Processing in Medical Imaging, (pp. 562–573). Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Velioglu, B., & Vural, F. T. Y. (2017). Transfer learning for brain decoding using deep architectures. In *ICCI*CC*, (pp. 65–70). IEEE.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*, 3371–3408.

Vu, A. T., Auerbach, E., Lenglet, C., Moeller, S., Sotiropoulos, S. N., Jbabdi, S., Andersson, J., Yacoub, E., & Ugurbil, K. (2015). High resolution whole brain diffusion imaging at 7 T for the Human Connectome Project. *NeuroImage*, *122*, 318–331.

Wachinger, C., Reuter, M., Initiative, A. D. N., et al. (2016). Domain adaptation for Alzheimer's disease diagnostics. *NeuroImage*, *139*, 470–479.

Wang, J., Chen, Y., Hao, S., Feng, W., & Shen, Z. (2017). Balanced distribution adaptation for transfer learning. In *2017 IEEE International Conference on Data Mining (ICDM)*, (pp. 1129–1134). IEEE.

Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., & Yu, P. S. (2018). Visual domain adaptation with manifold embedded distribution alignment. In *ACMMM*, (pp. 402–410). ACM.

Worsley, K. J., & Friston, K. J. (1995). Analysis of fmri time-series revisited—again. *Neuroimage*, *2*(3), 173–181.

Xiao, M., & Guo, Y. (2015). Feature space independent semi-supervised domain adaptation via kernel matching. *TPAMI*, *37*(1), 54–66.

Xue, G., Aron, A. R., & Poldrack, R. A. (2008). Common neural substrates for inhibition of spoken and manual responses. *Cerebral Cortex*, *18*(8), 1923–1932.

Yan, K., Kou, L., & Zhang, D. (2018). Learning domain-invariant subspace using domain features and independence maximization. *IEEE Transactions on Cybernetics*, *48*(1), 288–299.

Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, (pp. 188–197). ACM.

Yarkoni, T., Poldrack, R. A., Nichols, T. E., Van Essen, D. C., & Wager, T. D. (2011). Large-scale automated synthesis of human functional neuroimaging data. *Nature methods*, *8*(8), 665–670.

Yeo, B. T., Krienen, F. M., Sepulcre, J., Sabuncu, M. R., Lashkari, D., Hollinshead, M., Roffman, J. L., Smoller, J. W., Zöllei, L., Polimeni, J. R., et al. (2011). The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, (pp. 1125–1165).

Yokoi, S., Kobayashi, S., Fukumizu, K., Suzuki, J., & Inui, K. (2018). Pointwise hsic: A linear-time kernelized co-occurrence norm for sparse linguistic expressions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (pp. 1763–1775).

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49–67.

Zhang, C., Qiao, K., Wang, L., Tong, L., Hu, G., Zhang, R.-Y., & Yan, B. (2019). A visual encoding model based on deep neural networks and transfer learning for brain activity measured by functional magnetic resonance imaging. *Journal of Neuroscience Methods*, (p. 108318).

Zhang, H., Chen, P.-H., & Ramadge, P. (2018a). Transfer learning on fMRI datasets. In *AISTATS*, (pp. 595–603). PMLR.

Zhang, J., Li, W., & Ogunbona, P. (2017). Joint geometrical and statistical alignment for visual domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, (pp. 5150–5158). IEEE.

Zhang, Q., Filippi, S., Gretton, A., & Sejdinovic, D. (2018b). Large-scale kernel methods for independence testing. *Statistics and Computing*, *28*(1), 113–130.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., & Gordon, G. J. (2018). Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, (pp. 8559–8570).

Zhou, G., Cichocki, A., Zhang, Y., & Mandic, D. P. (2015). Group component analysis for multiblock data: Common and individual feature extraction. *IEEE transactions on neural networks and learning systems*, *27*(11), 2426–2439.

Zhu, Y., Zhuang, F., & Wang, D. (2019). Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, (pp. 5989–5996).

Appendices

Appendix A

Proofs

Lemma A.1 (Hoeffding's inequality (Hoeffding, 1963))**.** *Let $X_1, \ldots, X_m$ be independent random variables with X_i, taking values in $[a_i, b_i]$ for all $i \in [m]$. Then, for any $\epsilon > 0$, the following inequalities hold for $S_m = \sum_{i=1}^{m} X_i$:*

$$\mathbb{P}[S_m - \mathbb{E}[S_m] \geq \epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^{m}(b_i - a_i)}, \tag{A.1}$$

$$\mathbb{P}[S_m - \mathbb{E}[S_m] \leq -\epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^{m}(b_i - a_i)}. \tag{A.2}$$

Theorem 2.5 (Growth function generalisation bound (Vapnik & Chervonenkis, 1971))**.** *Given a hypothesis set $\mathcal{H}$, for $m \in \mathbb{N}$, $\epsilon \in (0, 1)$, for any $h \in \mathcal{H}$*

$$\mathrm{R}(h) \leq \hat{\mathrm{R}}(h) + \sqrt{\frac{8}{m} \ln \frac{4\Pi_{\mathcal{H}}(2m)}{\delta}}. \tag{2.6}$$

Proof. By Hoeffding Inequality in Lemma A.1 and Definition 2.4, the following probabilistic bound can be derived

$$\mathbb{P}\big[\big|\mathrm{R}(h) - \hat{\mathrm{R}}(h)\big| > \epsilon\big] \leq 4\Pi_{\mathcal{H}}(2m) \exp\big(-\frac{m\epsilon^2}{8}\big). \tag{A.3}$$

Denoting the right hand side of Eq. (A.3) as δ, i.e.

$$\begin{aligned} 4\Pi_{\mathcal{H}}(2m)\exp\big(-\frac{m\epsilon^2}{8}\big) &= \delta \\ \exp\big(-\frac{m\epsilon^2}{8}\big) &= \frac{\delta}{4\Pi_{\mathcal{H}}(2m)} \\ \frac{m\epsilon^2}{8} &= ln\frac{4\Pi_{\mathcal{H}}(2m)}{\delta} \\ \epsilon &= \sqrt{\frac{8}{m}\ln\frac{4\Pi_{\mathcal{H}}(2m)}{\delta}}, \end{aligned} \tag{A.4}$$

then Eq. (A.3) becomes

$$\begin{aligned} \mathbb{P}\big[\big|\mathrm{R}(h)-\hat{\mathrm{R}}(h)\big|>\epsilon\big] &\leq \delta \\ 1-\mathbb{P}\big[\big|\mathrm{R}(h)-\hat{\mathrm{R}}(h)\big|>\epsilon\big] &\geq 1-\delta \\ \mathbb{P}\big[\big|\mathrm{R}(h)-\hat{\mathrm{R}}(h)\big|\leq\epsilon\big] &\geq 1-\delta. \end{aligned} \tag{A.5}$$

We can say, with the confidence of $1-\delta$

$$\big|\mathrm{R}(h)-\hat{\mathrm{R}}(h)\big|\leq\epsilon \rightarrow \mathrm{R}(h)\leq\hat{\mathrm{R}}(h)+\epsilon, \tag{A.6}$$

then by substituting ϵ in terms of δ in Eq. (A.4), resulting in

$$\mathrm{R}(h)\leq\hat{\mathrm{R}}(h)+\sqrt{\frac{8}{m}\ln\frac{4\Pi_{\mathcal{H}}(2m)}{\delta}}. \tag{2.6}$$

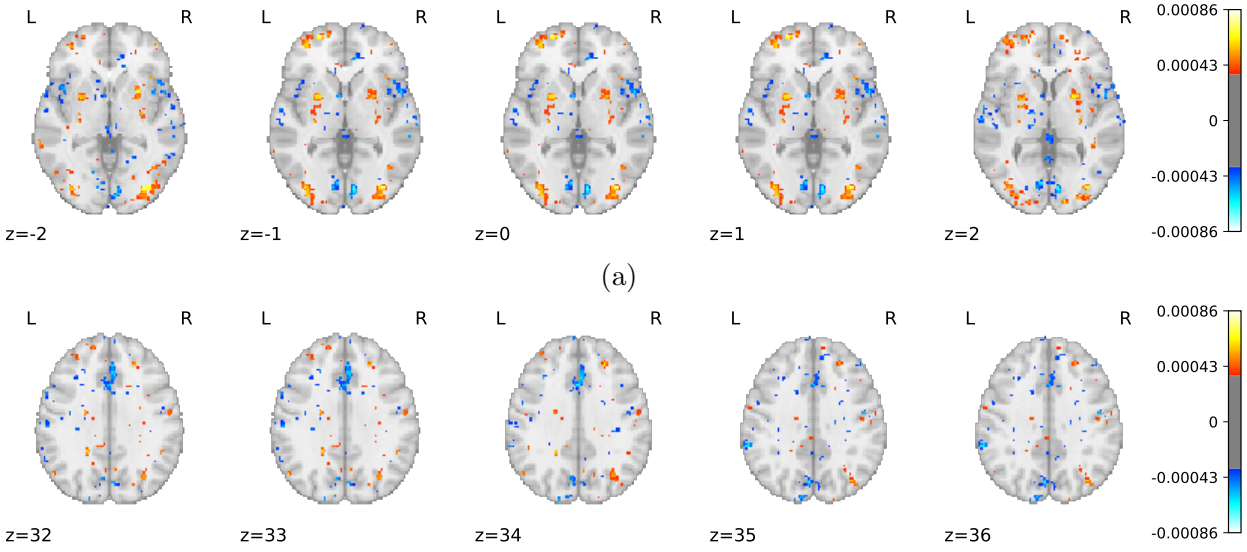
This completes the proof. □

Appendix B

Additional Experimental Results

B.1 Multi-Domain Brain Decoding

Additional Views of Learnt Activation Areas to Stop-Signal Tasks



(a)

(b)

Figure B.1: Activation areas identified by CoIR_{LS} across five cognitive experiments. The images are presented in the order from bottom (smaller value of z) to top (larger value of z). The figures are generated by the Python library Nilearn (Abraham et al., 2014).

Table B.1: Classification accuracy in percentage semi-supervised domain adaptation of brain decoding tasks on the selected datasets in Chapter 5 with same subjects. SVM and PCA (followed by a SVM) are trained on labelled target samples only. Avg denotes the averaged accuracy on the multi-source adaptation tasks.

	SVM	**PCA**	**TCA**	**ARSVM**	**MEDA**	**MIDA**	**CoIR$_{\text{SVM}}$**
A→B	63.4±2.6	62.5±4.4	55.0±4.3	65.7±2.2	57.8±5.3	64.5±5.0	**78.6±2.9**
B→A	59.2±4.0	60.4±5.7	58.9±3.2	64.0±2.8	62.5±4.5	71.2±3.5	**79.6±2.7**
A→C	68.4±2.6	66.9±6.3	70.6±6.3	70.9±2.7	70.1±2.1	78.4±2.9	**87.4±2.1**
C→A	59.2±4.0	60.4±5.7	63.6±4.0	59.5±3.4	59.7±3.1	**70.6±4.0**	68.8±3.8
B→C	68.4±2.6	66.9±6.3	86.4±5.3	73.5±2.1	73.6±3.6	80.1±4.4	**90.5±1.5**
C→B	63.4±2.6	62.5±4.4	75.3±4.0	62.5±2.1	57.0±2.1	73.2±3.3	**77.4±2.8**
B&C→A	59.2±4.0	60.4±5.7	52.1±1.1	63.2±2.5	52.6±1.4	51.4±1.8	**80.3±3.1**
A&C→B	63.4±2.6	62.5±4.4	54.5±3.1	67.6±1.6	52.6±1.3	61.7±2.2	**79.5±2.0**
A&B→C	68.4±2.6	66.9±6.3	52.1±1.1	71.9±2.6	57.0±0.8	65.4±2.2	**89.9±1.7**
Avg	63.7±3.1	62.6±5.5	52.9±1.8	67.6±2.2	54.1±2.6	59.5±2.0	**83.2±2.3**

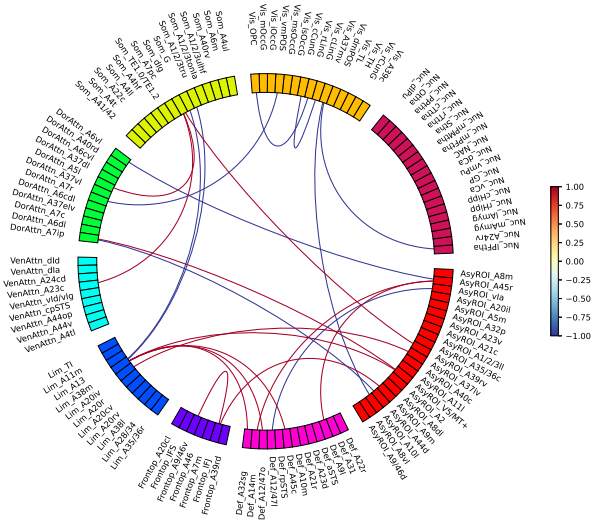
Multi-Source Adaptation Results under Semi-Supervised Setting

Table B.1 reports the brain decoding accuracy under semi-supervised domain adaptation setting (with labelled target domain samples). We have two key observations:

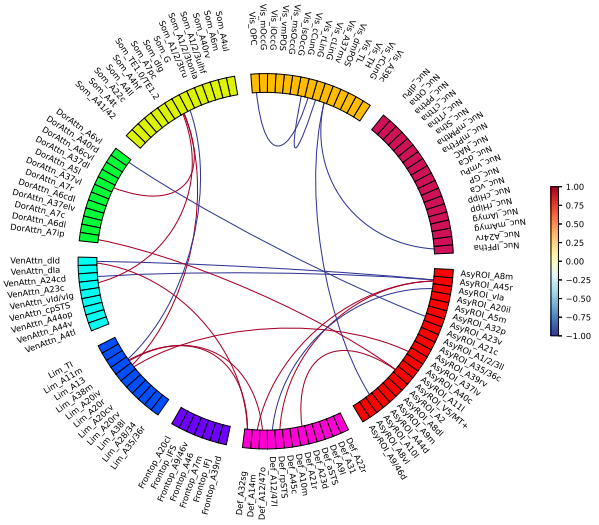
- On the whole, CoIR$_{\text{SVM}}$ outperformed all the comparing methods. CoIR$_{\text{SVM}}$ outperformed the best existing method (ARSVM) by **15.6%** (83.2% vs. 67.6%) in DA tasks with multi-source experiments. On the other hand, CoIR$_{\text{SVM}}$ obtained lower accuracy on A&B→C compared to B→C, and the rest results show using multiple source experiments is better. Thus, source selections can influence the transfer performance. If there is no clear preference of a particular source dataset, transfer with multiple sources is preferred in our opinion.

- MIDA and CoIR$_{\text{SVM}}$ outperformed the corresponding MMD-counterparts TCA and ARSVM, respectively. Based on Theorem 4.6, this confirmed that making use of multiple domain covariates (experiments and subjects) is beneficial in brain decoding.

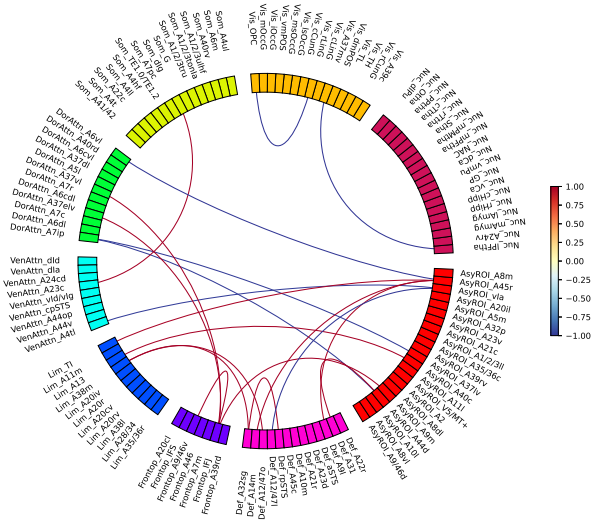
B.2 Gender-Related Brain Lateralisation



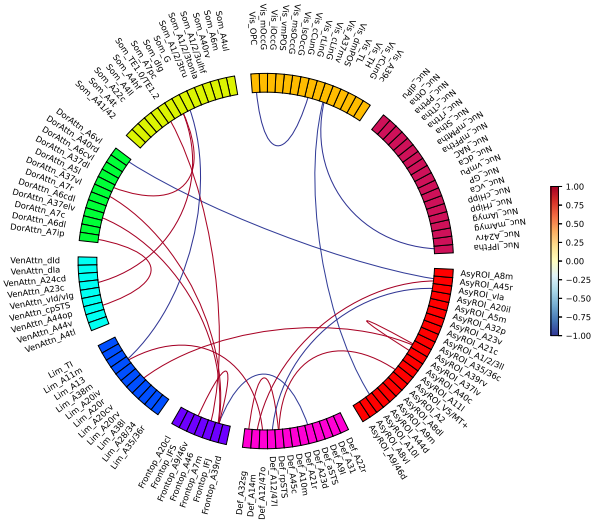
(a) Top general discriminative features of models trained on female samples when $\lambda = 0$



(b) Top general discriminative features of models trained on male samples when $\lambda = 0$

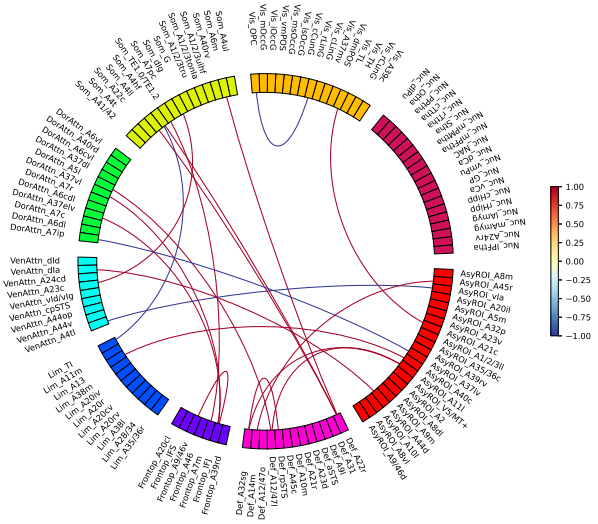


(c) Top female-specific discriminative features learnt when $\lambda = 1$

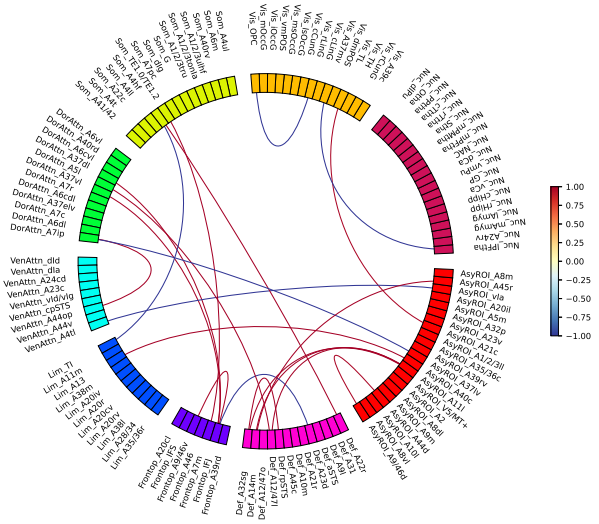


(d) Top male-specific discriminative features learnt when $\lambda = 1$

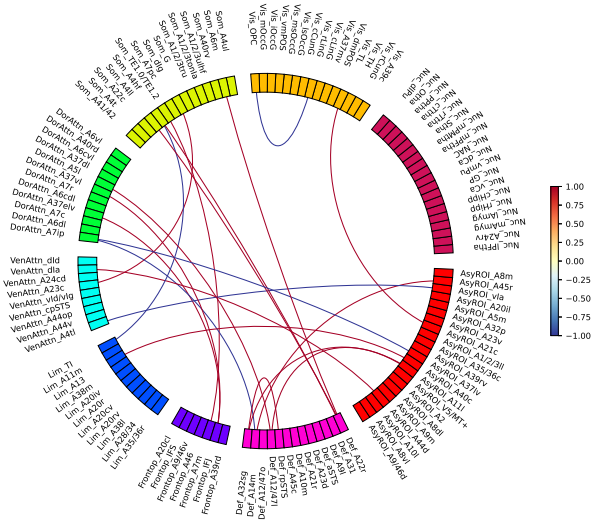
Figure B.2: Frequency of top discriminative features for left / right brain hemisphere classification among learnt specific models on ROIs of Yeo BrainNet R. The features were sorted by the weight magnitudes and the frequency over random train test splits.



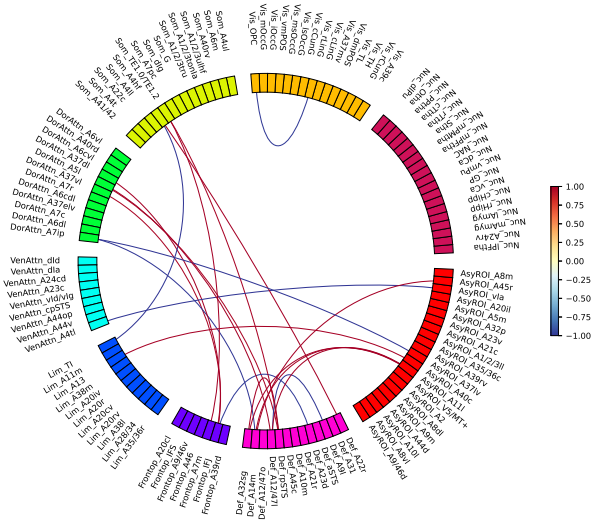
(e) Top female-specific discriminative features learnt when $\lambda = 2$



(f) Top male-specific discriminative features learnt when $\lambda = 2$

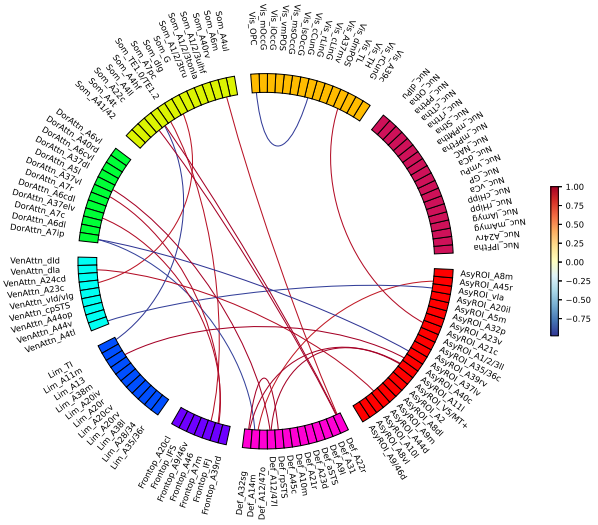


(g) Top female-specific discriminative features learnt when $\lambda = 5$

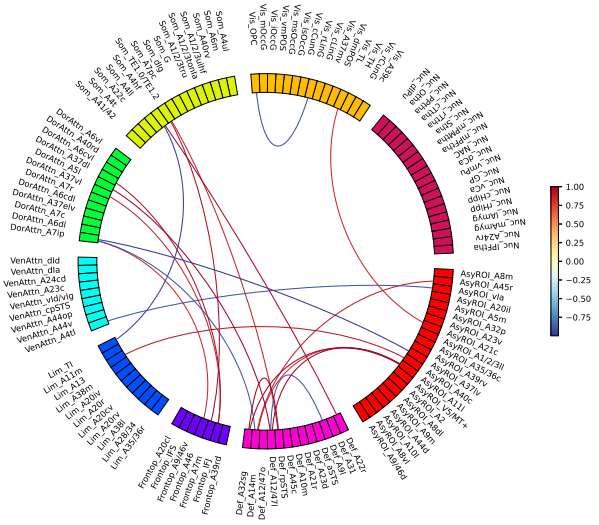


(h) Top male-specific discriminative features learnt when $\lambda = 5$

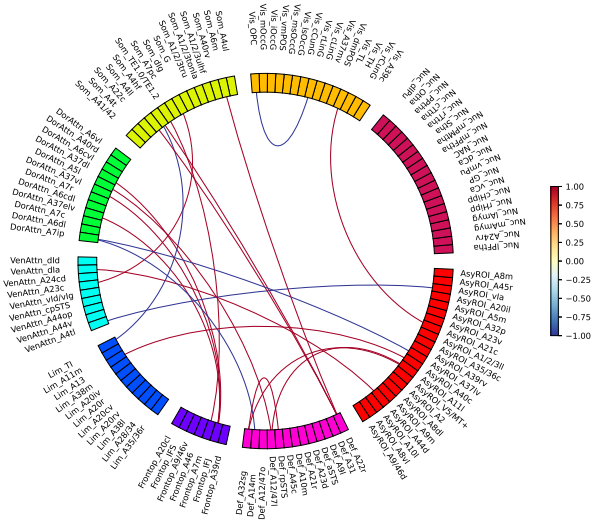
Figure B.2: Frequency of top discriminative features for left / right brain hemisphere classification among learnt specific models on ROIs of Yeo BrainNet R. The features were sorted by the weight magnitudes and the frequency over random train test splits.



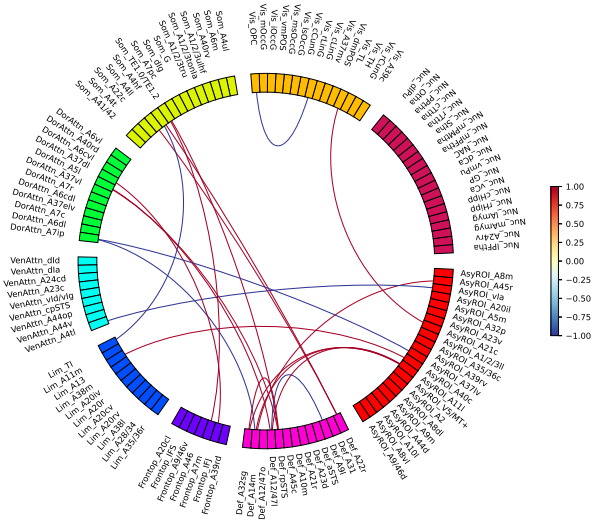
(i) Top female-specific discriminative features learnt when $\lambda = 8$



(j) Top male-specific discriminative features learnt when $\lambda = 8$



(k) Top female-specific discriminative features learnt when $\lambda = 10$



(l) Top male-specific discriminative features learnt when $\lambda = 10$

Figure B.2: Frequency of top discriminative features for left / right brain hemisphere classification among learnt specific models on ROIs of Yeo BrainNet R. The features were sorted by the weight magnitudes and the frequency over random train test splits.