Computational Neuroscience with Deep Learning for Brain Imaging Analysis and Behaviour Classification

Harrison Nguyen

A thesis submitted to fulfil requirements for the degree of:

Doctor of Philosophy (PhD)

Supervisor:
Dr. Fabio Ramos

The University of Sydney
School of Computer Science

2021

Declaration

I, *Harrison Nguyen*, declare that this thesis is submitted in fulfilment of the requirements for the conferral of the degree *Doctor of Philosophy (PhD)*, from the University of Sydney, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Harrison Nguyen

December 27, 2021

Abstract

The brain is a structure of mass and protein being composed up of two overarching types of cells, called glial and neurons, and it contains many billions of each. Neurons are known for gathering and transmitting electrochemical signals and the glial cells, on the other hand, provide physical protection to neurons and help keep them, and the brain, healthy. Whilst simple in its base components, this complex network of cells gives rise our thoughts, actions and emotions. Unfortunately, like every other organ in the human body, it is prone to damage, degeneration and disease.

Recent advances of artificial neural networks and deep learning model have produced significant results in problems related to neuroscience. For example, deep learning models have demonstrated superior performance in non-linear, multivariate pattern classification problems such as Alzheimer's disease classification, brain lesion segmentation, skull stripping and brain age prediction [1–5]. Deep learning provides unique advantages for high-dimensional data such as MRI data, since it does not require extensive feature engineering. The thesis investigates three problems related to neuroscience and discuss solutions to those scenarios.

Magnetic Resonance Imaging (MRI) has been used to analyse the structure of the brain and its pathology. However due to the heterogeneity of these scanners, MRI protocol, variation in site thermal and power stability, as well as site differences in gradient linearity, centring and eddy currents can introduce scanning differences and artefacts for the same individual undergoing different scans. Therefore combining images from different sites or even different days can introduce biases that

obscure the signal of interest or can produce results that could be driven by these differences. An algorithm, the CycleGAN, is presented and analysed which uses generative adversarial networks to transform a set of images from a given MRI site into images with characteristics of a different MRI site. Its purpose is to correct for differences in site artefacts without the need for *a priori* calibration using phantoms or significant coordination of acquisition parameters.

Secondly, the MRI scans of the brain can come in the form of different modalities such as T1-weighted and Fluid Attenuated Inversion Recovery (FLAIR) which have been used to investigate a wide range of neurological disorders. Current state-of-the-art models for brain tissue segmentation and disease classification require multiple modalities for training and inference. However, the acquisition of all of these modalities are expensive, time-consuming, inconvenient and the required modalities are often not available. As a result, these datasets contain large amounts of *unpaired* data, where examples in the dataset do not contain all modalities. On the other hand, there is a smaller fraction of examples that contain all modalities (*paired* data) and furthermore, each modality is high dimensional when compared to the number of data points.

This thesis presents a method to address the issue of translating between two neuroimaging modalities with a dataset of *unpaired* and *paired*, in semi-supervised learning framework. The proposed model, Semi-Supervised Adversarial CycleGAN (SSA-CGAN), uses an adversarial loss to learn from *unpaired* data points, cycle loss to enforce consistent reconstructions of the mappings and another adversarial loss to take advantage of *paired* data points. The experiments demonstrate that the proposed framework produces an improvement in reconstruction error and reduced variance for the pairwise translation of multiple modalities and is more robust to thermal noise when compared to existing methods.

Lastly, behavioural modelling will be considered, where it is associated with an impressive range of decision-making tasks that are designed to index sub-components

of psychological and neural computations that are distinct across groups of people, including people with an underlying disease. However, although the choices that various groups of patients may differ in apparently systematic ways, using this discrepancy to cleave to the original indexation has proved to be challenging. Current approaches either adopt complex discriminative models, essentially sacrificing interpretability, or use traditional computational models and/or manually chosen summary statistics at the expense of accuracy and scalability. The thesis proposes a method that learns prototypical behaviours of each population in the form of readily interpretable, subsequences of choices, and classifies subjects by finding signatures of these prototypes in their behaviour. The method extends recent suggestions for how the flexibility of recurrent neural networks can be combined with the interpretability of prototypes. The power of the method is illustrated on synthetic and real-world datasets, showing directly that we do not need to sacrifice accuracy for interpretability.

Acknowledgements

I would like to thank my supervisor Fabio Ramos for guiding me through this portion of my life. Without him, I would be a lost sheep in a dry desert hoping for some miracle to befall onto me. I would like to acknowledge and express my gratitude to Amir Dezfouli who gave me so much advice, knowledge and wisdom for the final stretch of my degree. To all my friends at the lab and CTDS, thank you. This experience would not been bearable without your friendship, lunches, drinks at the pub. Furthermore, this work would not have been produced without the work of people who are incredibly much smarter than me. I am indeed standing on the shoulders of giants.

Most importantly, I want to express my gratitude and dedicate this work to my parents. They have provided me a home, food, education without asking anything in return but despite all their sacrifice, both have unfortunately (and perhaps ironically for me) have succumbed to diseases of the brain and mind over this period. I hope my work will help us make one small step towards a better understanding of the brain and mind and eventually help those suffering from these diseases in the future.

Publications

H. Nguyen, R. W. Morris, A. W. Harris, M. S. Korgoankar, and F. Ramos. "Correcting differences in multi-site neuroimaging data using Generative Adversarial Networks." *arXiv preprint arXiv:1803.09375*, 2018.

H. Nguyen, S. Luo, and F. Ramos. "Semi-supervised Learning Approach to Generate Neuroimaging Modalities with Adversarial Training." *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham*, 2020.

Contents

List of Figures

List of Tables

Nomenclature

Background

$\mathbf{b}_1$ Bias of the 1^{st} neural network layer.$\mathbf{b}_i$ is the bias of the i^{th} layer.

$\mathbf{\Phi}(\mathbf{x})$ Parameterised basis functions to create new features from input $\mathbf{x}$

$\mathbf{W}_1$ Weights of the 1^{st} neural network layer. $\mathbf{W}_i$ is the weights of the i^{th} layer.

$\mathbf{x}_i$ the $\imath^{th}$ data point

y_i the label of the $\imath^{th}$ data point

Correction of MRI Multisite Differences

β Parameters of the regression model

G_θ A mapping function parameterised by θ

$\hat{P}(\mathbf{X})$ An estimate of the probability distribution for the imaging set $\mathbf{X}$

$P(\mathbf{X})$ The probability distribution for the imaging set $\mathbf{X}$

$\mathbf{X}$ A matrix or a set of images.

$\mathbf{x}$ A vector or an image

$\{\mathbf{y}_j\}_{j=1}^{M} \in \mathbf{Y}$ A set of M images

Semi-Supervised Imputation of Missing Modalities

D_X A function that discriminates between real images from X and fake/generated images of X

F A function that maps $Y \to X$

$\{x_i\}_{i=1}^N$ A set of images

Interpretable Modelling for Neuropsychological Tasks

atten The attention network which outpus a probability vector $\mathbf{z}_i$ which will determine the starting index of the subsequence $\tilde{\mathbf{x}}_i$

C The number of prototypes to discover for each class

$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ A dataset of N labeled data points

$\mathbf{d}_i$ A vector of the dissimilarity between the the encoding of the subsequence, $\tilde{\mathbf{x}}_i$ and the prototypes

$\mathbf{enc}(\theta_e, .)$ The recurrent neural network used to encode input sequence $\mathbf{x}_i$ with parameters θ_e

$\mathbf{M}$ The set of encoded prototypes $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_C\}$

$\mathbf{subenc}(\theta_{\mathbf{s}}, .)$ The encoder network for the subsequence $\tilde{\mathbf{x}}_i$

$\mathbf{u}_i^t$ The encoded vector of $\mathbf{x}_i^t$ by $\mathbf{enc}(\theta_e, .)$

$\mathbf{v}_i$ The encoding for the subsequence $\tilde{\mathbf{x}}_i$

$\mathbf{W}_f$ The weights of the final linear layer of the network

$\mathbf{x}_i = (\mathbf{x}_i^t)_{t=1}^{T_i}$ A sequence of length T_i. Each element of the sequence is a d-dimensional real vector for the i-th individual which includes the reward and action of subject i at time step t

$\tilde{\mathbf{x}}_i = (\mathbf{x}_i^t)_{t=k_i}^{k_i+L-1}$ A subsequence of $\mathbf{x}_i$ of length L that begins at index k_i

Abbreviation

AC-PC Anterior-posterior commissure. 55

Adam Adaptive Moment Estimation. 22, 61, 86

ADC Apparent diffusion coefficient. xiv, 78

BOLD Blood Oxygen Level Dependent. 6

BraTS Brain Tumor Segmentation Dataset. xiii, xiv, 6, 9, 11, 40, 77–79, 84, 85, 119

CBF cerebral blood flow. xiv, 78, 85, 88, 89

CBV cerebral blood volume. xiv, 78, 85, 88

cGAN conditional GAN. 31, 80

cLR-GAN Conditional Latent Regressor GAN. 32

CNN Convolutional Neural Network. 17, 19, 61

CSF cerebrospinal fluid. 43, 44, 55

CT Computerised Tomography. xiv, 6, 78, 81

cVAE-GAN Conditional Variational Autoencoder GAN. 32

CWRG Cycle Wasserstein Regression GAN. 79, 86, 88, 89, 91

CycleGAN Cycle Generative Adversarial Network. iv, xiii, 11, 49, 56, 79–83, 86, 88, 89, 91, 117, 118

DCNN Deep Convolutional Neural Network. 77, 79, 82

DSM Diagnostic and Statistical Manual of Mental Disorders. 53

DWI Diffusion Weighted Imaging. 42, 77

f-MRI Functional Magnetic Resonance Imaging. 6, 95

FLAIR Fluid Attenuated Inversion Recovery. iv, xiii, xiv, 40, 42, 44, 77, 78, 84, 88

GAN Generative Adversarial Network. xiv, xvii, 9, 28–30, 53–55, 57–59, 61–63, 66, 67, 69–75, 79, 80, 82, 120

GM grey matter. 55, 62–64, 75

GP Gaussian Process. 62, 63, 69–73, 75

GRU Gated Recurrent Unit. xviii, 20, 21, 100, 104, 111

HGG High Grade Glioma. 84

ICU Intensive Care Unit. 79

IR inversion recovery. 44

ISLES Ischaemic Stroke Lesion Segmentation Dataset. xiv, xviii, 6, 9, 11, 77–79, 84, 85, 87, 119

KL Kullback-Leibler. 27, 96

LGG Low Grade Glioma. 84

LSGAN Least Squares GAN. 30, 57

LSTM Long Short Term Memory. 20–22

MAE Mean Absolute Error. 87–89

MB model-based. 109, 110, 112

MDP Markov Decision Process. 33, 35

MF model-free. 109, 110, 112

MLE Maximum likelihood Estimation. 28, 29

MNI Montreal Neurological Institute brain standard. 55

MR Magnetic Resonance. 7, 9, 11, 49, 51–53, 81, 85, 86, 88, 118

MRI Magnetic Resonance Imaging. iii, iv, 6, 7, 9, 11, 40, 42, 49–54, 66, 72, 73, 75, 77, 79, 81, 84, 85, 92, 117, 120

MSE Mean Squared Error. 16, 73, 87–89

MTT mean transit time. xiv, 78, 85, 88

MVPA Multi-Voxel Pattern Analysis. 74

OCD obsessive compulsive disorder. 93

OCT Optical coherence tomography. 81

PCA Principal component analysis. xiv, 25, 26, 64, 69, 70

QLG Q-learning greedy. 107

QLR Q-learning *repeated*. 107, 108

ReLU Rectified Linear Unit. 22, 60, 61

RF radio frequency. 41–44

RGB Red-Green-Blue. 17

RL Reinforcement Learning. xiii, 33, 46, 102

RMSProp Root Mean Square Propagation. 22, 106

RNN Recurrent Neural Network. 19–22, 94, 95, 97, 104, 108, 112, 115

SSA-CGAN Semi-Supervised Adversarial CycleGAN. iv, 9, 79, 83, 86–89, 92, 119

SSA-CGAN-p SSA-CGAN trained with only paired examples. 87, 88

SSL semi-supervised learning. 80

SVM Support Vector Machine. 64, 65, 67, 70–72

t-SNE T-distributed Stochastic Neighbour Embedding. xiv, 26, 27, 69, 70

T1 T1-weighted. xiii, xiv, 40, 42–44, 78, 84, 85

T1c T1 with gadolinium enhancing contrast. xiii, xiv, 40, 42, 43, 78, 84, 85, 91

T2 T2-weighted. xiii, xiv, 40, 42, 44, 78, 84, 85, 88, 91

TD Temporal Difference. 34–36, 46–48

TE Time to Echo. 42–44

TI time to inversion. 45

Tmax time-to-max. 85

TR Repetition Time. 42–44

TTP time-to-peak. xiv, 78, 85

VBM Voxel-Based Morphometry. 74

WAIS Wechsler Adult Intelligence Scale. 53

WM white matter. 55

Chapter 1

Introduction

1.1 Motivation

1.1.1 The Brain

The brain is considered the most complex part of the human body. Whilst being 1.3kg in weight, or about 2% of the average human's body weight, it is the centre of the nervous system which interprets our external senses, the initiator of body movement and most importantly, the creator of our thoughts and intellect. With 10^{11} cells called neurons, several hundred trillion synaptic connections, the brain has massive parallel processing capacity allowing us to comprehend images in 100ms [6].

When completing a maths problem, for example, it is subconsciously processing data from millions of nerve cells that handle the visual input of the paper, the sensory input from the tactical and aural senses, and combines this data to keep track of the position of the paper and pen. Whilst at the same time, the brain retrieves from memory past experiences related to the problem at hand, regulates our heartbeat, controls our hormones, manages our hunger and thirst. Due to the complex responsibilities of this organ and the heavy metabolic demands of brain cells, 20% of the blood pumped is received by the brain [7]. Unfortunately, like every other organ in the human body, it is prone to damage, degeneration and

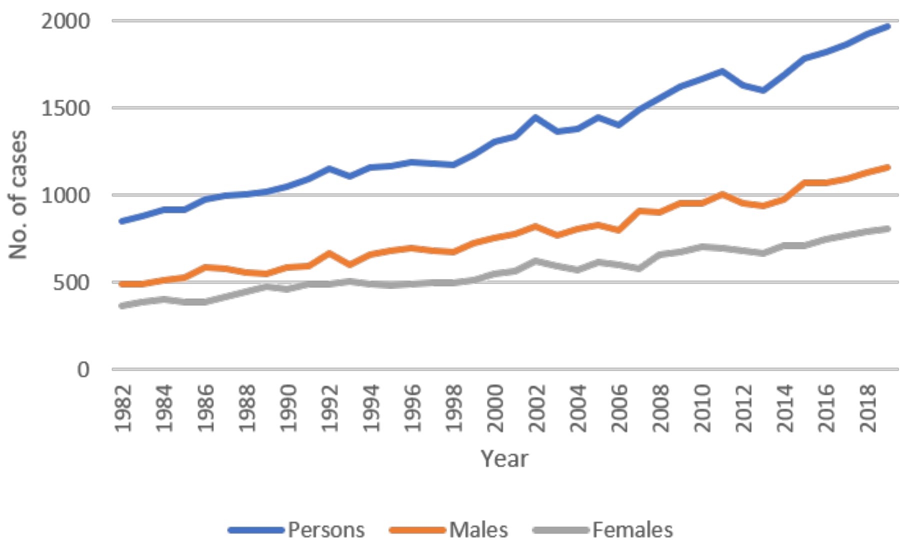


Figure 1.1: Incidence of brain cancer in Australia from 1982-2019, by sex [8].

disease.

For instance, benign tumours can form in the brain which are slow growing and unlikely to spread to other parts of the body. On the other hand, malignant brain tumours are incredibly cancerous and able to spread into other parts of the brain or spinal cord, impacting brain functions such as memory areas, speech and language, perceptual and reasoning function. In Australia, brain cancer incidence has increased by 130% from 1982 until 2019 (in comparison the Australian population has increased by 64.6% over that period of time) and was the 10th most common cause of cancer death in 2018 with the age-standardised incidence rate of 6.7 cases per 100,000 persons in 2016.

However the prognosis for those with primary brain cancer is not promising. For those diagnosed with other cancers, individuals had a 70% chance of surviving five years compared to their counterparts in the general population. On the other hand, for those diagnosed with primary brain cancer, between 1987-1991, the relative five year survival rate was 20.6% and despite all the significant progress in science and technology, this had improved to 22.2% between 2012-2016; a 7.8% improvement almost 30 years later [8].

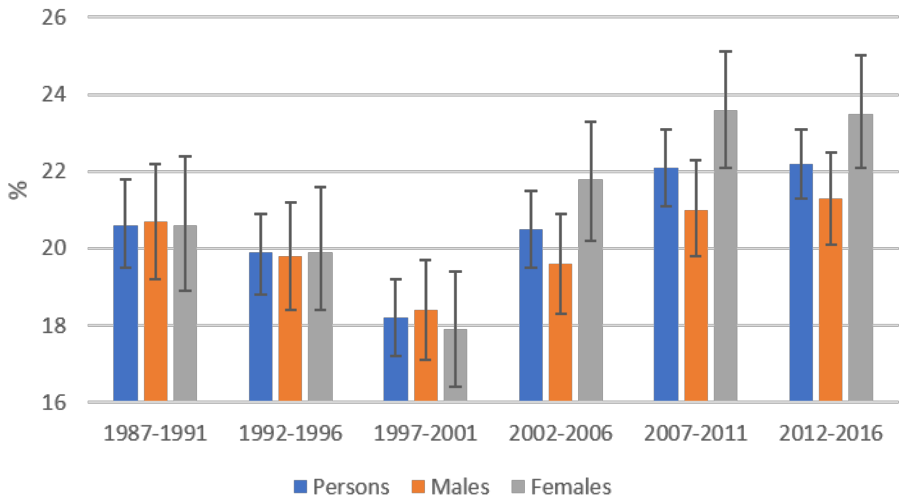


Figure 1.2: 5-year relative survival rate of brain cancer in Australia 1987–1991 to 2012–2016, by sex [9].

1.1.2 The Mind

While the brain's cells and circuitry can be seen as the equivalent to the hardware of the computer, we also have the human equivalent of software: the mind which processes mental representations, meaning, emotions, and judgement.[1] However, much like how a software contains bugs, mental illness is prevalent in the population. In Australia, mental illness affects one in five people in the age of 16-85, of which, depression and anxiety and substance-use disorder is the most common [10]. Furthermore, these illness tend to occur together, increasing the chance of detrimental outcomes.

Prevalence rates vary across the lifespan but are highest in the early adult years-the period during which people are usually establishing families and independent working lives (Figure 1.3). Typically the experience of mental illness during this period has its onset in childhood or adolescence and can have long term implica-

[1]Some researchers studying mental illnesses believe that abnormalities in the brain circuits' function, through chemical neurotransmitters, contribute to the development of many mental illnesses. These connections between the nerve cells can lead to problems with how the brain processes information and may result in abnormal mood, thinking perception or behaviour. The relationship between the mind and to the physical brain and nervous system is still an open problem.

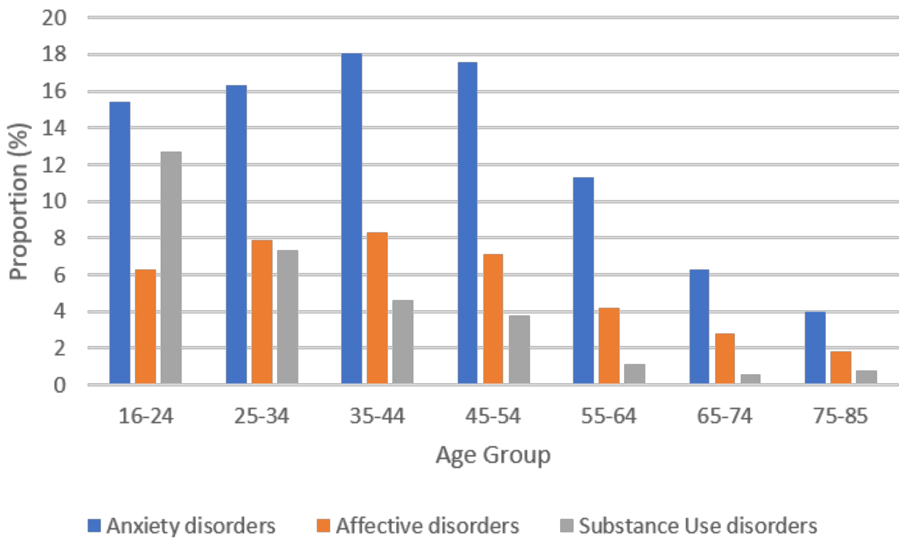


Figure 1.3: Prevalence of mental illness by age group. Affective disorders are mood disorders e.g. depression. A person may have had more than one mental disorder [15].

tions. Its impact on people's well-being is so significant that it is the largest single cause of disability in Australia, accounting for 24% of the burden of non-fatal disease and is associated with the lowest likelihood of being in the labour force with all of its impacts costing $60 billion to the Australian economy [11, 12]. Furthermore, the percentage of men and women being diagnosed and treated with these disorders has increased (Figure 1.5,1.3). This poses a conundrum, however, where although there is greater public visibility of the importance of mental health and more people seeking treatment, there has not been a decrease in the prevalence of these illnesses but rather, has remained remained steady between 2001-2018 [13, 14] (Figure Figure 1.4).

The physiological mechanics and causes for mental illnesses is still debated in the literature but its understanding can be divided into two approaches: the biological understanding of psychiatric problems and the psychological framework. The biologically driven framework understands the nature of mental illness through the lens of biochemical imbalances, genetic factors and pathophysiolgy of the brain. In contrast, the psychological perspective focuses on information processing and the mind

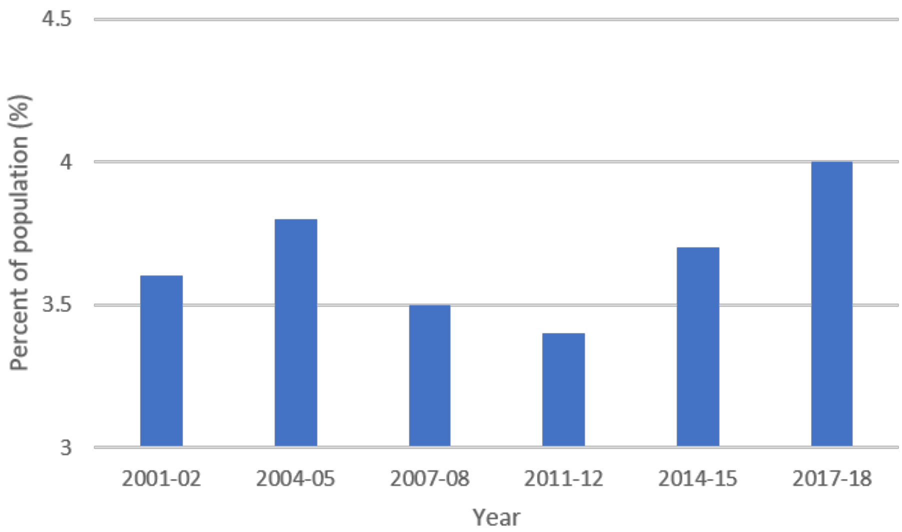


Figure 1.4: The percentage of the population reporting very high Kessler Psychological Distress Scale (K10) score, where a score between 30 and 50 is considered very high, over the previous month, from 2001-02 to 2017-18. This is used as a proxy of the levels of depression and anxiety symptoms in the population [15].

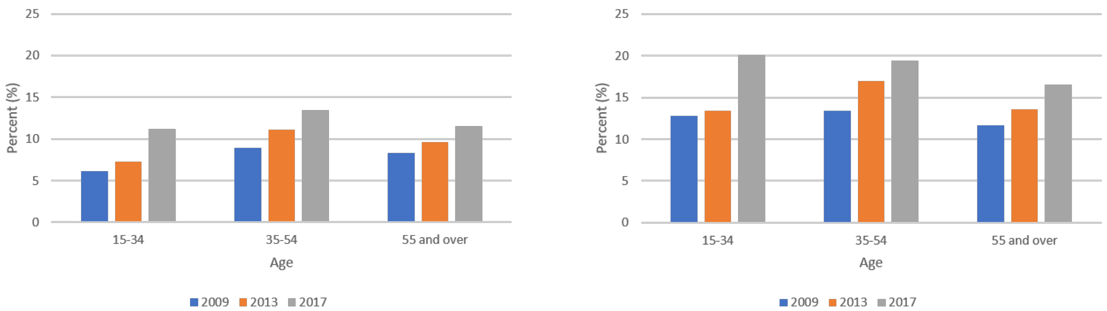


Figure 1.5: Percentage of men and women respectively, diagnosed with depression or anxiety, by age group; 2009, 2013 and 2017 [16].

and events such as psychological reactions to stressors, negative beliefs, meaning and feedback loops [17].

1.1.3 Understanding the Brain and Mind

With recent advances in technology, the medical and neuroscience community has an expanding set of tools to diagnose, investigate and understand the causes for brain diseases and mental illnesses. For example, neurocognitive assessments provides a detailed assessment of all major functions of the brain, such as storage and retrieval of memory, expressive and receptive language abilities, calculation, dexterity, and the overall well-being of the patient. Neuropsychological computational tasks such as the bandit task or the two-stage task is able to identify behavioural differences between patients with bipolar disorder and healthy patients [18].

MRI and Computerised Tomography (CT) scans are used to measure the size of brain tumours and determine their severity. Functional Magnetic Resonance Imaging (f-MRI) is able to describe the pattern of brain connectivity among different regions by looking at changes in the neuronal blood flow known as the Blood Oxygen Level Dependent (BOLD) signal. Applications of the f-MRI showed that when asked to identify the emotions displayed on a series of facial images presented supraliminally or subliminally, patients with schizophrenia showed reduced activity in the right amygdala and medial prefrontal cortex during conscious perception of fear when compared to healthy controls [19]. Resting state f-MRI, which measures spontaneous low-frequency fluctuations in the BOLD signal, is a relatively new pathway for evaluating regional interactions in the absence of tasks where the persistent level of background activity of the brain during rest, called the default mode network, has been applied to the study of many brain diseases and mental illnesses such as Alzheimer's disease, schizophrenia and bipolar disorder [20]. The richness of data of the brain is epitomised by large projects and datasets such as ADNI [21], BraTS [22], ISLES [23], OpenfMRI [24] which attempt to collate and share data.

These expanding datasets create a primate environment for data driven methods to solve various problems in this domain and broaden the scope of neuroscientific research and insight.

In particular, due to their ability to learn complex feature representations from raw data, deep learning models have had success in many domains in neuroscience such as tumour segmentation [22], Alzheimer's disease classification [25, 26], analysing neural connectivity [27, 28] and risk prognostication [29]. This thesis will extend the applications of deep learning models in various potential problems in the neuroscience domain which are described in Section 1.2.1.

1.2 Deep Learning Applications in Neuroscience

1.2.1 Challenges

The success of deep learning methods has stemmed partially from the large volumes of data available. However in the neuroscience domain, datasets that are composed of Magnetic Resonance (MR) images are relatively limited in the number of samples and the dimensionality of the data is much larger than the number of examples i.e. $D >> N$.

Furthermore, due the heterogeneity of MR scanners, MRI protocol, variation in site thermal and power stability, as well as site differences in gradient linearity, centring and eddy currents can introduce scanning differences and artefacts for the same individual undergoing different scans. Combining images from different sites or even different days can introduce biases that obscure the signal of interest or can produce results that could be driven by these differences. This can make the interpretation, reliability and reproducibility of findings difficult. Thus, methods need to be developed that can correct these multi-site differences in order to pool data which provides the opportunity to address a major source of concern regarding the low statistical power of published studies, especially when larger studies are not feasible due to financial constraints or recruitment is difficult [30].

Many state-of-the-art machine learning models in brain tissue segmentation and disease classification require multiple modalities during training and inference. However, examples where all modalities are available is limited and therefore the ability to incorporate examples that do not use all but some modalities could be important for the adoption of these methods in clinical settings or to improve existing models. In other words, scenario being presented is a case when the dataset has paired examples (examples with all modalities available) and unpaired examples (examples with only one available modality).

Although deep learning models perform well in metrics such as classification accuracy, it is an incomplete description of most real-world tasks. In certain tasks such as movie recommendations where the result may not have significant consequences or when the problem is sufficiently well-studied, the former may be reasonable. However, when there is an incompleteness in the problem formalisation, it is not sufficient to predict the outcome of an event but the model should explain *how* the prediction was determined [31]. For example, testing the scenarios in which a system may fail for complex tasks may be computationally or logistically impossible but by being able to explain the behaviours of the system, provides insight to possible edge cases or inputs that cause the system to fail. Especially in clinical settings, where diagnosis of a disease can have serious consequences, deep learning models need to be designed to be interpretable in order to properly evaluate the model outside of metrics such as accuracy before deployment and to increase the trust between the system and users [32].

1.3 Contributions

1.3.1 Unsupervised Correction of MRI Multi-site Differences

An algorithm will be presented and analysed that uses Generative Adversarial Network (GAN) [33] to transform a set of images from a given MRI site into images with characteristics of a different MRI site. Its purpose is to correct for differences in site artefacts without the need for *a priori* calibration using phantoms or significant coordination of acquisition parameters. This algorithm can be treated as a 'black box' without knowledge of the artefacts present in the dataset and can be applied *post hoc* after acquisition to two unpaired sets of imaging data. Importantly, as demonstrated by the results, the correction occurs without any apparent loss of information related to gender or clinical diagnosis.

1.3.2 Semi-supervised Imputation of Missing MR Modalities

The thesis investigates particular context where there are two sets of MR volumes in the dataset: 1) a set where each example contains all available modalities, 2) a set where has at least one missing modality of which forms the majority of the dataset. In this work, a method is developed to address these issues with semi-supervised learning in translating between two neuroimaging modalities. The proposed model, Semi-Supervised Adversarial CycleGAN (SSA-CGAN), uses an adversarial loss to learn from the former examples, cycle loss to enforce consistent reconstructions of the mappings and another adversarial loss to take advantage of latter examples. This method can be used to input missing modalities in the dataset as well as translate unseen MR volumes between modalities. The method is evaluated on two datasets, BraTS [34] and ISLES [23] which have been used to evaluate state-of-the-art methods for segmentation of brain tumours and lesions. The experiments demonstrate that the proposed framework produces an improvement in reconstruction error and reduced variance for the pairwise translation of multiple modalities and is more robust to thermal noise when compared to existing methods.

1.3.3 Interpretable modelling for Neuropsychological Tasks

This thesis describes a novel framework which learns a prototype subsequence for each group of subject to characterise their overall behaviour, and also learns to extract a (smaller) subsequence from the behavioural data of each individual which can be assessed against group prototypes. The model takes the behavioural data of a set of individuals as input and learns to classify them by comparing the similarity of the individual subsequences with group prototypes which are learned simultaneously. The framework, therefore, can be used to classify subjects into groups in an interpretable way by finding 'witness' subsequences in the behaviour of each individual. Furthermore, is also able to extract subsequences from each group which exemplifies the whole group's behaviour. Through a set of experiments, we show that in terms of classification performance the proposed method is similar to current methods while offering interpretability. The framework is validated using synthetic data and also shows that when applied to the behaviour recorded from healthy and patients with bipolar disorders, the model is able to extract the signature behaviours of each group. The framework therefore, offers a novel method for behavioural data analysis and may find applications in different areas of behavioural analytic, decision-neuroscience and computational psychiatry.

1.4 Outline

The main material of the thesis begins with Chapter 2 which describes the relevant background material for the work. Section 2.1 explains neural networks such as feed-forward neural networks, convolutional neural networks, recurrent neural networks and then deep neural networks. Section 2.2 describes unsupervised learning techniques and in particular generative adversarial networks. Section 2.3 delves into another framework in machine learning, reinforcement learning and describe two techniques to solve the problem. Section 2.4 provides an overview of illnesses of the mind and brain and the tools used to study them, such as Magnetic Resonance Imaging (MRI) and psychological computational tasks.

Chapter 3 discusses the application of deep learning using a particular deep learning model, the CycleGAN [35], to correct for multi-site differences in MR scans and demonstrates how the algorithm improves upon existing methods with respect to predictive capabilities. Section 3.2 describes previous methods that were used to create predictive models in the presence of confounders such as scanner differences. Section 3.3 describes the method used and the deep learning architecture used for this problem. Section 3.4 describes five experiments that demonstrate that this method improves upon existing methods in terms of classification and reconstruction performance.

Chapter 4 investigates a novel algorithm for semi-supervised domain translation of MR images where instead of using only *unpaired* examples of imaging sets as in Chapter 3, the framework leverages *paired* examples to improve reconstruction error. The problem is introduced in Section 4.1 and related work is discussed in Section 4.2. Section 4.3 revisits the CycleGAN but also describes a further modification of the CycleGAN to create a novel framework that leverages existing *paired* data by including an additional adversarial loss. Section 4.4 demonstrates the performance of this method on multiple datasets, ISLES [23] and BraTS [34] and its improvement in terms of reconstruction error and robustness in the presence of noise.

Chapter 5 presents a novel framework for an interpretable deep learning model used in understanding neuropsychological tasks. The context of the problem is presented in Section 5.1 and discusses the importance of including interpretability in designing models for high stakes decision making such as diagnosis of disease. Section 5.3 presents the architecture of the proposed model and the algorithm used to train the model. Section 5.4 compares the performance of the method against other models and analyses the performance of the model under different conditions such as the length of the sequence used for interpretation and ablation studies. Section 5.5 discuses the results of the experiments, limitations of the method and potential future work.

Chapter 6 concludes the thesis by providing a summary of the work discussed in the thesis in Section 6.1 and provides some ideas for future work in Section 6.2 .

Chapter 2

Background

In this chapter, the background of the ideas presented in this thesis is introduced. First, neural networks and their variations are described in Section 2.1. Section 2.1.4 discusses deep learning architectures and how they are used to model different phenomena. Then Sections 2.2 and 2.3 give an overview of some of other classes of machine learning problems; unsupervised and reinforcement learning respectively. Finally, the background will conclude with Section 2.4 by describing some of the potential diseases of the mind and some of the tools used to understand them.

2.1 Neural networks

Suppose we have a dataset given by N input-output pairs,

$$\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_i, y_i), ..., (\mathbf{x}_n, y_n)\},$$

where $\mathbf{x}_i \in \mathrm{R}^D$ and y_i is a scalar, for simplicity. If we assume that there exists a *linear* mapping between each $\mathbf{x}_i$ and y_i (with potentially, y_i being augmented with noise), our model in this case, is a linear transformation of the inputs $f(\mathbf{x}) = \mathbf{x}\mathbf{W} + b$ with $\mathbf{W}$ is some $D \times 1$ matrix over the reals and b a scalar. Different parameters, $\mathbf{W}$ and b, define different linear transformations and the aim is to find the parameters that minimise an *objective function*, which for example, could be minimising the average squared error over the observed data, $\frac{1}{N}\sum_i(y_i - \mathbf{x}\mathbf{W} - b)^2$.

While some observations can be defined by a linear function, e.g. the conversion between Celsius and Fahrenheit, or can be approximated by one, e.g. the relationship between voltage and current, however, in general, the relationship between $\mathbf{x}_i$ and y_i does not need to be linear and we may wish instead define a non-linear function mapping, $f(\mathbf{x})$, between inputs and outputs.

We can model these relationships using parameterised basis functions, $\mathbf{\Phi}(\mathbf{x}) = [\phi_1^{\mathbf{w}_1,b_1}, ..., \phi_k^{\mathbf{w}_K,b_K}]$, where $\phi_k^{\mathbf{w}_k,b_k}$ is a scalar valued function of the inner product $\langle \mathbf{w}_k, \mathbf{x}\rangle + b_k$ [36]. In other words, we transform input $\mathbf{x}$ using K fixed, scalar valued nonlinear functions to produce a feature vector, $\mathbf{\Phi}(\mathbf{x})$. For example, $\phi_k(\cdot) = \tanh(\cdot)$ giving $\phi_k^{\mathbf{w}_k,b} = \tanh(\langle \mathbf{w}_k, \mathbf{x}_k\rangle + b_k)$. Typically, each ϕ_k is the same basis function. The feature vector produced by the output of the basis functions can be fed as an input to another linear transformation. Written more compactly, $f(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})\mathbf{W}_2 + b_2$ where $\mathbf{W}_2$ is a matrix of dimension $K \times 1$, b_2 a scalar, $\mathbf{\Phi}(\mathbf{x}) = \phi(\mathbf{x}\mathbf{W}_1 + \mathbf{b})$ with $\mathbf{W}_1$ a $D \times K$ matrix and $\mathbf{b_1}$ is a vector of K elements. Again, we minimise an objective function, $\frac{1}{N}\sum_i(y_i - f(\mathbf{x}_i))^2$, to find these parameters $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$ and b_2.

Hierarchies of these parameterised basis functions can be formed, $\mathbf{\Phi}(\mathbf{x})$, through their composition, e.g. $f(\mathbf{x}) = \mathbf{\Phi}_2(\mathbf{\Phi}_1(\mathbf{x}))\mathbf{W}_3 + b_3$, creating what is known as a *neural network*. Each composition is known as a *layer* of the neural network and forms the building block of *deep learning* models, where many layers are composed to form a model that can capture more complex functions.

The following sections will delve into particular deep learning models, beginning with feed-forward neural networks, through to ones that can process image and sequence data.

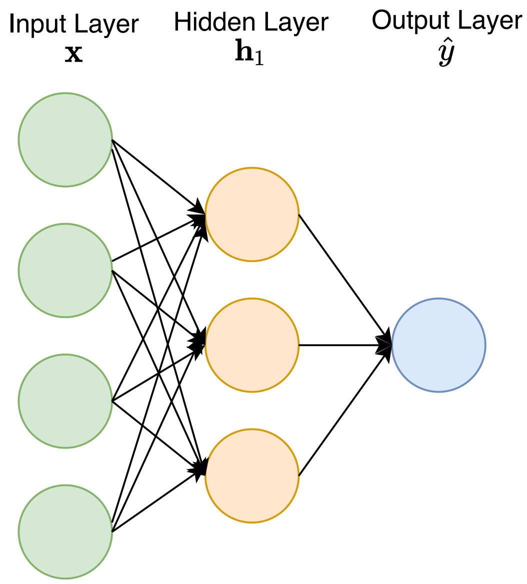


Figure 2.1: The architecture of a 1 hidden layer feed forward neural network.

2.1.1 Feed-forward Neural Networks

We begin by reviewing the feed-forward neural network using a *single hidden layer* for simplicity and ease of notation. We denote the model input, $\mathbf{x} \in \mathrm{R}^D$, as the input layer. The input undergoes a linear transformation, $\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1$, where $\mathbf{W}_1$ is known as the *weight* matrix of size $D \times K$, and $\mathbf{b}_1$, the bias being a vector of K elements. An element-wise non-linearity is applied to the linear transformation to produce the *hidden layer*,

$$\mathbf{h}_1 = \sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1), \tag{2.1}$$

where $\sigma(\cdot)$ is known as the *activation function*. Examples of activation functions are the sigmoid function $\sigma(x) = \frac{1}{1+\exp^{-x}}$ or the ReLU, $\sigma(x) = \max(0, x)$.

The hidden layer undergoes another transformation to be mapped to the output of the model or the *output layer*,

$$\hat{y} = \mathbf{h}_1\mathbf{W}_2 + b_2, \tag{2.2}$$

where $\mathbf{W}_2$ is a weight matrix of $K \times 1$ and b_2 a scalar.

Putting these equation together, we get

$$\hat{y} = \sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + b_2. \tag{2.3}$$

The parameters of the network, $\mathbf{W}_1$, $\mathbf{W}_2$, b_2 and $\mathbf{b}_1$ is estimated from training data by minimising the parameters w.r.t a loss function. In regression problems, an example could be the Mean Squared Error (MSE),

$$\mathcal{L}_\theta(\mathbf{X}, \mathbf{y}) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{2.4}$$

where θ are the parameters of the model, $\mathbf{y}$ are the N observed outputs.

In classification problems, where we instead predict the probability of $\mathbf{x}$ being classified with a label of the set $\{0, 1\}$ (binary classification), the model output $\hat{y}$, is passed through sigmoid (logistic) function, $\hat{p}_i = \frac{1}{1+\exp^{-\hat{y}_i}}$. The loss of the network can be calculated as

$$\mathcal{L}_\theta(\mathbf{X}, \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_1)). \tag{2.5}$$

The parameters of the model is estimated using gradient descent which calculates the gradient of the loss function, $\mathcal{L}_\theta(\mathbf{X}, \mathbf{y})$ with respect to the parameters, θ. Then according to a specified learning rate, α, each iteration of the gradient descent updates the weights and biases according to

$$\theta^{t+1} = \theta^t - \alpha \frac{\delta \mathcal{L}_{\theta^t}(\mathbf{X}, \mathbf{y})}{\delta \theta}, \tag{2.6}$$

where θ^t denotes the parameters of the neural network at iteration t of the gradient descent algorithm.

We can extend the ideas of the above single hidden layer neural network with

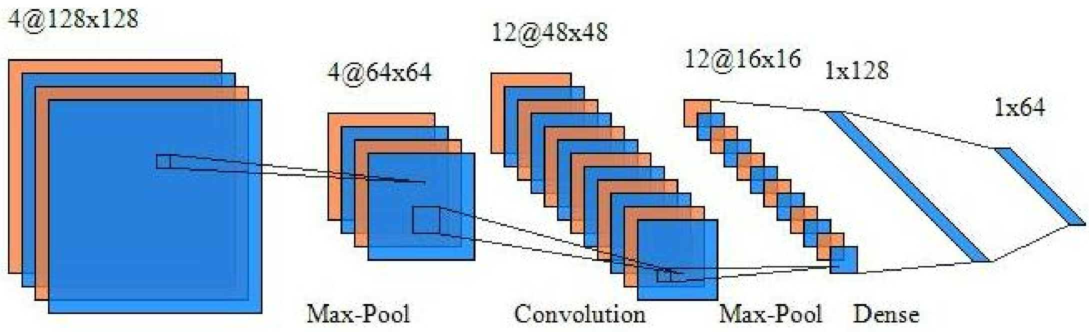


Figure 2.2: A convolutional neural network with two sets of convolution operations, along with max pool operations and two fully connected (dense) layers to predict a label with 64 classes.

multiple hidden layers (called *deep learning*) to create a more expressive model that can capture more complex relationships between inputs and outputs. Certain structures can also be designed such that they aimed at using particular inputs such as images and sequence. The following sections will review these models.

2.1.2 Convolutional Neural Networks

The Convolutional Neural Network (CNN) is a popular deep learning tool for image processing. The hidden layers of the model typically consists of a series of *convolutional* layers that perform a sliding dot product (despite being known as convolutions) that preserves spatial information of the image input.

The input of a convolutional layer is a tensor that has a given height, width and depth (e.g. RGB channels). After being passed through a convolutional layer, the image is transformed to a *feature map* with shape, (feature map height) x (feature map width) x (feature map channels) which are determined by the *convolutional kernel* (See Figure 2.3). The width and height of the convolutional kernel dictates the receptive field in which the layer processes the input and the number of *filters* (feature map channels) in the kernel has been thought of being feature identifiers e.g. the filters on the first layer are used to detect edges, simple colours and curves [37].

To perform the operation, the kernel, $\mathbf{k}$, of size $m \times m$ is placed over a selected

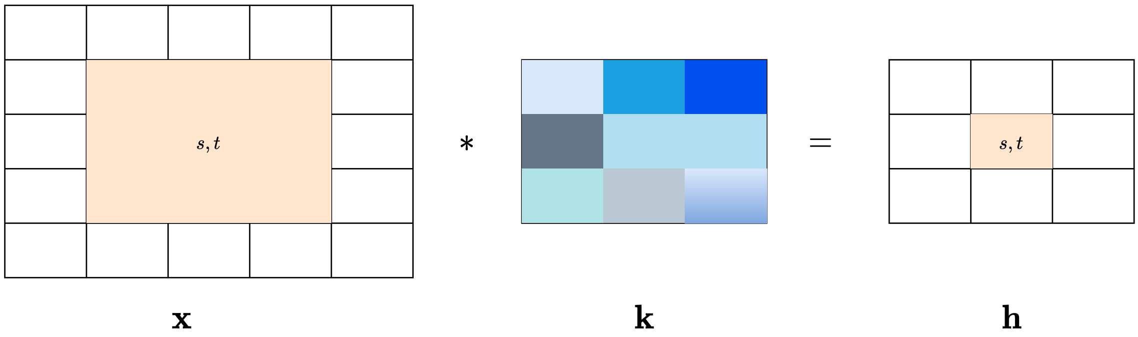


Figure 2.3: An example of a convolution operation showing the receptive field of a kernel.

pixel. Each of the values from the kernel is multiplied with the corresponding values from the image that are under the receptive field of the kernel. This result is summed to produce an element in the feature map. This process is repeated, by shifting the kernel by a specified amount, known as the *stride*, (e.g. 1 pixel) to complete the feature map.

More mathematically, given some 2D input with height and width but no depth (for simplicity), $\mathbf{x} \in \mathbb{R}^{H \times W}$, and a convolutional layer with a kernel, $\mathbf{k}$, of size $m \times m$ and thus has m^2 parameters, the output of the the layer, $\mathbf{h}$, will be the size, $(H - m + 1) \times (W - m + 1)$. The value given for the row and column of the output with index i and j is given by:

$$\mathbf{h}[i, j] = (\mathbf{x} * \mathbf{k})[i, j] = \sum_{s} \sum_{t} \mathbf{k}[s, t]\mathbf{x}[i - s, j - t]. \tag{2.7}$$

Typically, convolutional layers are interspersed with *pooling* layers such as *max pooling* or *average pooling* that calculates the maximum or average value respectively, for each patch on the feature map. The result of using a pooling layer and creating down sampled or pooled feature maps is to form a summarised version of the features detected in the input. This allows the network to be invariant to local translation or, in other words, when the input is translated by a small amount, the values of most of the pooled outputs do not change.

While a fully connected feed-forward network can be used to learn features for

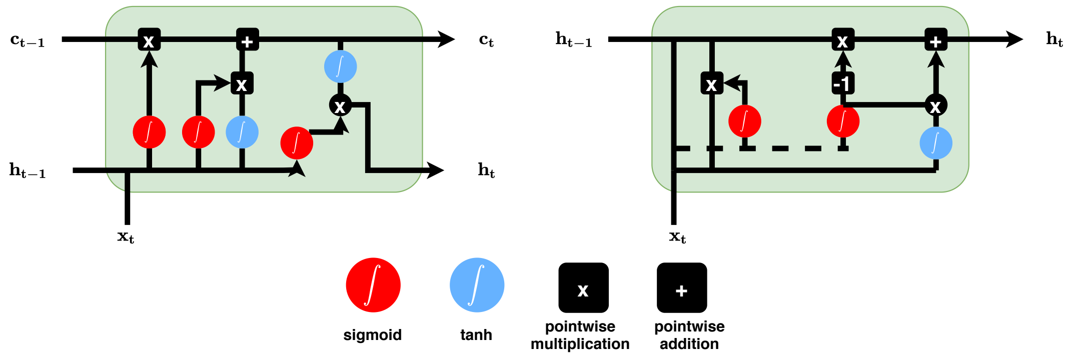


Figure 2.4: Diagram and operations of LSTM (left) and GRU (right) cells.

image data, a large number of parameters would be necessary for shallow (one hidden layer) architectures, due to the very large input sizes associated with images, where each pixel is a relevant variable. The advantage of using convolutional layers is that it reduces the number of free parameters allowing for *deeper* neural network architectures and thus more expressive models.

2.1.3 Recurrent Neural Networks

While CNNs are used for processing image inputs, Recurrent Neural Network (RNN) are sequence-based models used in tasks such as machine translation [38], speech recognition [39] and generating image descriptions [40]. Given input sequence $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_T]$ of length T, a function $\mathbf{f_h}$ is applied to each time step, $\mathbf{x}_t$, to produce a hidden state, $\mathbf{h}_t$. This newly generated, $\mathbf{h}_t$, is rolled forward along with $\mathbf{x}_{t+1}$ to create a new hidden state, $\mathbf{h}_{t+1}$.

$$\mathbf{h}_{t+1} = \mathbf{f_h}(\mathbf{x}_{t+1}, \mathbf{h}_t) = \sigma(\mathbf{x}_{t+1}\mathbf{W_h} + \mathbf{h}_t\mathbf{U_h} + \mathbf{b_h}), \text{ for } t = 0, 1, ..., T-1, \tag{2.8}$$

where $\mathbf{h}_0 = 0$, $\sigma(\cdot)$ is the activation function and $\mathbf{W_h}, \mathbf{U_h}, \mathbf{b_h}$ are parameters to be learned.

The output of the model uses the last hidden state, $\mathbf{h}_T$, and a linear transformation

$$\hat{y} = \mathbf{h}_T\mathbf{W}_y + b_y. \tag{2.9}$$

The intuition behind RNNs is that the information from earlier time steps is encoded in the hidden state, $\mathbf{h}_t$ and is passed forward and combined with future timesteps. However, due to the difficulties of learning the parameters of a RNN especially as the length of the input increases, more complex RNNs structures have been developed such as the Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Gated Recurrent Units

The GRU controls the flow of information being passed to future timesteps through two gating units called the reset, $\mathbf{r}$ and update, $\mathbf{z}$. Each gate depends on the previous hidden state, $\mathbf{h}_{t-1}$ and the current input, $\mathbf{x}_t$. A *cell* of a GRU can be described by the following equations:

$$\mathbf{z}_t = \sigma_g(\mathbf{W}_z\mathbf{x}_t + \mathbf{h}_{t-1}\mathbf{U}_z + \mathbf{b}_z), \tag{2.10}$$

$$\mathbf{r}_t = \sigma_g(\mathbf{W}_r\mathbf{x}_t + +\mathbf{h}_{t-1}\mathbf{U}_r + \mathbf{b}_r), \tag{2.11}$$

$$\hat{\mathbf{h}}_t = \phi_h(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{h}_{t-1} \odot \mathbf{r}_t) + \mathbf{b}_h), \tag{2.12}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \hat{\mathbf{h}}_t, \tag{2.13}$$

where $\odot$ is the Hadamard product, σ_g is the sigmoid function and ϕ_h is the hyberolic tangent function. Due to the sigmoid function, $\mathbf{z}_t$ and $\mathbf{r}_t$ are in the range $[0, 1]$ and thus act as gating functions. The update gate, $\mathbf{z}_t$, determines what information to remove from $\mathbf{h}_{t-1}$ and what new information to add from $\hat{\mathbf{h}}_t$. When the reset gate $\mathbf{r}_t$ is close to 0, the current hidden state is forced to ignore the previous hidden state and reset with the current input. $\mathbf{U}_z$, $\mathbf{U}_r$, $\mathbf{U}_h$, $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}_h$ are matrices to be learned. This allows the GRU to forget any past information that is not relevant to the future [41].

Long Short Term Memory

LSTM is an alternative RNN architecture. While the GRU has two gating units, the reset and update gates, the LSTM has three gating units, the input gate, $\mathbf{i}$, output gate, $\mathbf{o}$, and a forget gate, $\mathbf{f}$ and its operations can be described by the following operations:

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f\mathbf{x}_t + \mathbf{h}_{t-1}\mathbf{U}_f + \mathbf{b}_f), \tag{2.14}$$

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i\mathbf{x}_t + \mathbf{h}_{t-1}\mathbf{U}_i + \mathbf{b}_i), \tag{2.15}$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o\mathbf{x}_t + +\mathbf{h}_{t-1}\mathbf{U}_o + \mathbf{b}_o), \tag{2.16}$$

$$\tilde{\mathbf{c}}_t = \phi_h(\mathbf{W}_c\mathbf{x}_t + +\mathbf{h}_{t-1}\mathbf{U}_c + \mathbf{b}_c), \tag{2.17}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{2.18}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi_h(\mathbf{c}_t). \tag{2.19}$$

While the update gate in the GRU determines which information is removed and which information is added to the hidden state vector, this operation is separated in the LSTM using the forget gate, $\mathbf{f}$, and input gate, $\mathbf{i}$, respectively. In the GRU, the internal memory of the unit and the output vector of the unit is represented by the same vector, $\mathbf{h}_t$. However in the LSTM, this has been separated into two vectors, $\mathbf{c}_t$ and $\mathbf{h}_t$. The cell state, $\mathbf{c}_t$ acts as the internal "memory unit" of the LSTM, which is either added with new information from the input gate or is previous information forgotten through the forget gate. The output vector of the LSTM unit is the hidden state vector, $\mathbf{h}_t$, where the output gate $\mathbf{o}$, determines how much information is revealed from the cell state, $\mathbf{c}_t$. Again, $\mathbf{U}_f$, $\mathbf{U}_i$, $\mathbf{U}_o$, $\mathbf{U}_c$, $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$, $\mathbf{W}_c$ are matrices to be learned.

2.1.4 Deep Learning

While the previous sections discussed a single layer of a neural network, a *deep* networks can be formed through the composition of many hidden layers. For example, a deep learning network can be created by stacking multiple fully connected layers or multiple LSTM cells.

The parameters of a neural networks is trained, using a method called *backpropagation* (short for backward propagation of errors), where the gradients of the lower layers is calculated through a chained computation of the weight values of the layers above it. At first, the gradient of the final layer of weights is calculated and this calculation proceeds backwards through to the first layer of weights. Partial computations of the gradient from one layer are reused in the computation of the gradient for the previous layer. This backwards flow of the error information allows for efficient computation of the gradient at each layer [42].

Initially, the difficulty of extending a shallow network to one with more layers was due to the problem of vanishing gradients or exploding gradients [43]. For instance, in the vanishing gradient problem, the weight values of the layer above are very small, the gradients of the lowest layer will be exponentially smaller, reaching towards 0 and thus the weights of the lowest layer could stop changing its value and perhaps cause the whole network to stop learning.

This issue was alleviated through empirical experiments, using various techniques such as Xavier initialisation of the weight parameters [44], alternative optimisation methods such as Root Mean Square Propagation (RMSProp) [45] and Adaptive Moment Estimation (Adam) [46], alternative activation functions such as the Rectified Linear Unit (ReLU) [47] and in the case of RNNs, the use of LSTMs [48] instead.

The advantage of these architectures is that they are able to learn more complex functions, where layers further from the input layer are able to learn more complex features by incrementally building upon the features from previous layers. This allows deep networks to solve complex tasks without the need to manually engineer features (*feature engineering*) but instead learns task-specific features from raw data.

While deep networks are able to learn complex features, these models contain hundreds of thousands to millions of parameters that need to be estimated which makes them susceptible to *overfitting* where the model does not generalise well from the training data, which is used to estimate the parameters, to *unseen* data. To overcome this issue of overfitting, one method is that these models are often trained using large datasets. Some standard datasets, particularly for image-related problems include, ImageNet [49], CIFAR-10 [50] of which are often used to evaluate the performance of various deep learning architectures. The standardisation of datasets has also been filtered to neuroscience domain problems particularly for tasks such as tumour segmentation [34] and, Alzheimer's Disease classification [51].

However, the use of large datasets may still not alleviate the problem of poor generalisation to unseen data. Another technique that can be considered is the use of *regularisation*, where the size of the weights of the neural network is constrained. This is performed by including an extra term to the loss function during optimisation where it penalises the training of the model based on the magnitude of the weights. This encourages the model to learn to map the input to output whilst at the same time balancing to keep the network weights small [52].

Other techniques to reduce overfitting of neural networks which can be used in combination include:

1) activity regularisation- the magnitude of the activation of neurons is penalised [47],

2) weight constraint- the weights of the network are constrained to be below a specified magnitude [53],

3) dropout- neurons or weights are zeroed probabilitistically during training [54],

4) noise- the inputs of layers are corrupted with statistical noise during training [55],

5) early stopping- the performance of the model is monitored on a validation set and training is stopped when the performance degrades [56],

6) data augmentation- training data is for example flipped, rotated, enlarged to increase the number of examples the network is trained [57].

So far we have discussed deep learning models to be used in *supervised learning* problems where a function is learned that maps an input to an output based on example input-output pairs, i.e. learn $\mathbf{f} : \mathbf{X} \rightarrow \mathbf{Y}$. A training example is a pair that consists of an input and a desired output value. A supervised learning algorithm uses the training data to update the parameters of the model to infer the function, $\mathbf{f}$, which can be used to map new examples. The next section introduces other form of machine learning problems, namely unsupervised learning and reinforcement learning and certain methods that can be used to solve these problems.

2.2 Unsupervised Learning

In unsupervised learning, the training data consists of a set of input vectors $\mathbf{x}$ without any corresponding target values. The goal is to learn patterns in the data without any preexisting labels. There are several different goals in unsupervised learning such as clustering where the aim is to discover groups of similar examples within the data, or to project data from a high-dimensional space to two or three dimensions for the purpose of visualisation or to determine the distribution of data within input space, known as density estimation. In the latter case of density estimation, a model is trained to infer a priori distribution, $p(\mathbf{X})$, where $\mathbf{X}$ is the data matrix of size $N \times K$, where N is the number of observations and K is the number of features. In contrast to supervised learning, the algorithm intends to infer the conditional probability distribution $p(\mathbf{Y}|\mathbf{X})$, where each row in $\mathbf{Y}$ is the target vector for each row in $\mathbf{X}$.

The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization, or to determine the distribution of data within the input space, (i.e. learn $p_{data}(\mathbf{X})$), known as density estimation.

2.2.1 Principal Component Analysis

Principal component analysis (PCA) is an example technique for dimensionality reduction that focused on linear manifolds which is either a line, a plane or a hyperplane depending upon the number of dimensions involved [58]. These lower dimensions provide a succinct summary of the relationships between observed variables which are constructed by a number of linear transformations of those variables with certain optimality properties. PCA is defined as an orthogonal linear transformation that constructs the data to a new coordinate system such that some scalar projection of the data that lies on the first coordinate produces the greatest variance (called the principal component), the second coordinates produces the second greatest variance and so on.

More concisely, given a data matrix $\mathbf{X}$ of size $N \times K$, with column-wise empirical mean, where N is the number of observations and K is the number of features, we seek to find a transformation of weight coefficients, $\mathbf{w}_l = (w_1, w_2, ..., w_K)_l$ that maps each row vector $\mathbf{x}_i$ to a new vector $\mathbf{u}_i = (u_1, u_2, ..., u_L)_i$ given by

$$\mathbf{u}_i = \mathbf{x}_i \cdot \mathbf{w}_l, \tag{2.20}$$

where $l = 1, 2, ..., L$, L is the number of new vector transformations and $i = 1, 2, .., N$. Furthermore, $\mathbf{w}_l$ is constrained be unit length and each new variable as result of the transformation inherits the maximum possible variance from $\mathbf{X}$.

For example to find the first weight vector $\mathbf{w}_1$, we solve the following problem,

$$\begin{aligned}\mathbf{w}_1 &= \underset{||\mathbf{w}=1||}{\arg\max}\,\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i\cdot\mathbf{w}_1)^2\\ &= \underset{||\mathbf{w}=1||}{\arg\max}\,\frac{1}{N}\sum_{i=1}^{N}\mathbf{w}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w}\\ &= \underset{||\mathbf{w}=1||}{\arg\max}\,\mathbf{w}^T(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T)\mathbf{w}.\end{aligned}$$

Solving this gives the first principal eigenvector of the covariance matrix of the data $\Sigma = \mathbf{w}^T(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T)$. More generally, if we wish to project the data to a lower L-dimensional subspace, we choose $\mathbf{w}_1, ..., \mathbf{w}_L$ to be the top L eigenvectors of Σ.

The full principal components decomposition of $\mathbf{X}$ can therefore be given as

$$\mathbf{U} = \mathbf{X}\mathbf{W} \tag{2.21}$$

where $\mathbf{W}$ is a $K \times K$ matrix of weights whose columns are the eigenvectors of $\mathbf{X}^T\mathbf{X}$.

2.2.2 t-SNE

T-distributed Stochastic Neighbour Embedding (t-SNE) is another dimensionality reduction method and unlike PCA, is a nonlinear technique for embedding high-dimensional data for visualisation in a lower dimensional space [59].

t-SNE starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities where similar objected are assigned a high probability while dissimilar points are assigned a lower probability. More precisely, the similarity of datapoint $\mathbf{x}_j$ to $\mathbf{x}_i$, is the conditional probability, $p_{j|i}$, that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbour if neighbours were picked in proportion to a probability density centred as $\mathbf{x}_i$. This conditional probability is given as

$$p_{j|i} = \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-||\mathbf{x}_i - \mathbf{x}_k||^2/2\sigma_i^2)}. \tag{2.22}$$

Furthermore, $p_{j|i}$ is defined as

$$p_{j|i} = \frac{p_{j|i} + p_{i|j}}{2N}. \tag{2.23}$$

The bandwidth of the Gaussian kernels, σ_i, is adapted to the density of the data where smaller values of σ_i are used in denser parts of the data space.

Next, t-SNE defines a similar probability distributions over the points in the low-dimensional map, however instead of using a Gaussian distribution, it uses a distribution that has much heavier tails, which in this case is chosen to be a Student t-distribution with one degree of freedom. Using this distribution, the joint probabilities of the low dimensional map q_{ij} is given as

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}, \tag{2.24}$$

where $\mathbf{y}_i$ is $\mathbf{x}_i$ in the low dimensional map.

The location of the points $\mathbf{y}_i$ in the map is determined by minimising, using gradient descent, the Kullback-Leibler (KL) divergence of the distribution P from the distribution Q,

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{2.25}$$

2.2.3 Generative Adversarial Networks

Another task related in unsupervised learning is generative modelling which learns the patterns in input that in such a way that the model can be used to generate new examples that could have been drawn from the original dataset.

One advantage of studying these generative models is that they can be trained with missing data and can impute inputs that are missing. One particular case of missing data is *semi-supervised learning* where the most labels (but not all) of the training examples is missing. Particularly in cases where labelled examples are difficult to obtain, semi-supervised learning can be used to reduce the number of labels and instead learn to improve generalisation by using the more numerous unlabelled examples.

The Generative Adversarial Network (GAN) [33] is an example of a generative model that is capable of generating data from a given distribution, $p_{data}(\mathbf{X})$, as well able as to perform semi-supervised learning. The following section will describe the mechanics of GANs in the former scenario.

Before delving into the GAN framework, we first discuss the idea of *maximum likelihood*.

Maximum likelihood Estimation (MLE) [36] defines a model that providea an estimation of a probability distribution, parameterised by parameters θ. The *likelihood* function, measures the goodness-of-fit of the probability distribution to the training data for given values of the unknown parameters. It is defined as,

$$\prod_{n=1}^{N} p_{model}(\mathbf{x}_i; \theta), \tag{2.26}$$

where $\mathbf{x}_i$ is an example of the training set of size N. In MLE, the aim is to choose parameters of the model, θ, such that the likelihood is maximised,

$$\theta^* = \arg\max \prod_{n=1}^{N} p_{model}(\mathbf{x}_i; \theta). \tag{2.27}$$

Typically, the log-likelihood is maximised as it is less prone to numerical problems and since the logarithm is a strictly increasing function, maximising the log-likelihood is equivalent to maximising the likelihood,

$$\theta^* = \arg\max \sum_{n=1}^{N} \log p_{model}(\mathbf{x}_i; \theta). \tag{2.28}$$

The taxonomy of generative models that uses the principle of maximum likelihood can be divided into two main branches, explicit density models and implicit density models (of which GANs are a member).

In explicit density models, a density function is chosen, $p_{model}(\mathbf{x};\theta)$, and then the parameters of the model are estimated using MLE. Whilst the optimisation process is straightforward, the difficulty lies in designing a tractable model that can capture all of the complexity of the data generating process. As such, there are several strategies that can be used that either involve the construction of models that ensures tractability such as in fully visible belief networks [60] or models that make tractable approximations to likelihood as in the variational autoencoder [61].

Implicit density models, on the other hand, can be trained without explicitly defining a density function but instead train the model through sampling p_{model}. Examples include of implicit density model include the generative stochastic network which uses Markov chains but fails to scale to high dimensional spaces [62] and the Generative Adversarial Network (GAN) [33].

The GAN framework

While traditional machine learning models learn by minimising an objective functions, GANs succeed through the idea of adversarial training, where the model's training process can be described as a game between two players. One player is called the *generator* where it attempts to create samples from the same distribution as the observed data. The other player is the *discriminator* where its function is to examine the fake samples from the generator and real samples from the observed data and to classify the generated and observed samples as either real or fake [63].

Over time, the discriminator is trained with supervision to better distinguish real and fake samples. However at the same time, the generator will improve its synthesis of fake samples in order to fool the discriminator, which in turn will make the job of the discriminator more difficult. Eventually the solution of this game is a Nash equilibrium, where the generator is unable to improve its generation of fake samples and the discriminator is unable to better classify real and fake samples [63].

More formally, the *generator* is any differential function, G, (e.g.. a convolutional network) that takes as input $\mathbf{z}$ sampled from a prior distribution $p_{prior}(\mathbf{z})$ (e.g. a Gaussian distribution with diagonal covariance matrix) and transforms $\mathbf{z}$ into $G(\mathbf{z}) = \hat{\mathbf{x}}$ which is a sample from p_{model}.

The *discriminator* is similarly any differential function, D. It takes two mini-batches of data, the real samples from p_{data}, which are labelled as 1 and the fakes samples generated from p_{model}, labelled as 0. The parameters of D, θ^D, are updated through gradient based optimisation algorithms by minimising

$$\mathcal{L}_D = -\frac{1}{2}\mathrm{E}_{x\sim p_{data}}\big[\log D(\mathbf{x})\big] - \frac{1}{2}\mathrm{E}_{\mathbf{z}\sim p_{prior}}\big[\log(1 - D(G(\mathbf{z})))\big]. \tag{2.29}$$

$\mathcal{L}_D$ is the standard cross-entropy loss for binary classification models with a sigmoid output. In the original formulation of the GAN, the loss of the generator is tied directly to the discriminator loss whereby creating a zero-sum game,

$$\mathcal{L}_G = -\mathcal{L}_D. \tag{2.30}$$

From this point of view, the discriminator is more like a teacher instructing the generator in how to improve than an adversary. Alternative loss functions have been proposed for both the discriminator and generator, such as the LSGAN [64] which have improved training stability of GANs or improved the generated samples.

Image to Image Translation with GANs

There has been numerous applications of GANs in areas such as generating images from text [65], photographs of human faces [66], anime characters [67] or to increase the resolution of images [68]. These models have been used particularly in the problem of Image to image translation where the aim is to learn a mapping between images from a *source* domain to a *target* and even vice versa. For example, the model aims to learn to transfer the style from the source image to the target image, for example transforming a photograph into a artwork by Claude Monet. e Isola et al. [69] proposed the *pix2pix* model which used image pairs $\{A, B\}$, where A and B are two different depictions of the same underlying scene, and used a cGAN to learn the mapping between these paired images. To improve the resolution of the generated image, the generator used an encoder-decoder architecture with skip connections, much like the "U-net" [70] and had the discriminator to restrict its attention to the structure in local image patches. This was done by only penalising the discriminator at the scale of patches. Thus, instead of classifying an entire image as either real or fake, the discriminator would try to classify if each patch in an image as either real of fake.

Wang et al. [71] further improved the resolution of generated images to 2048×1024 by using a generator network that can be decomposed into two parts, the global generator network G_1 and the local enhance network G_2. G_1 was trained to generate images at a resolution of 1024×512 to create the global features of the image and G_2 focused on learning to improve the resolution of the image to 2048×1024. Conversely, three discriminators with similar architectures were used, D_1, D_2, D_3, but each operated on different image scales, where D_1 was trained on images at the original resolution, D_2 was trained on images that were downscaled by a factor of 2 and D_3 was trained on images that had a resolution 4 times smaller than the original resolution.

Both of these aforementioned models were studied with only considering translating between two image domains. The StarGAN [72] on the other hand, was proposed for multi-domain image-to-image translation using only a single generator and a discriminator. The generator was trained to translate an input image to an output image, conditioned on domain label information. To learn this conditional image mapping, an auxiliary classifier was added to the discriminator where the classification of the image domain of real images was used as loss to train the discriminator. On the other hand, the generator was trained by the domain classification of its generated images. Furthermore, in order to learn the mapping among many domains using only a single generator and discriminator, a mask vector was used to control the domain labels and to ignore unspecified labels.

So far these models assumed that there was a one-to-one mapping between the image domains. However what is more appropriate is that these image-to-image problems could be modelled as a single input image being able to correspond to multiple possible outputs. The BicycleGAN [73] attempted to capture the multi-modality of the output by using a low-dimensional latent vector. At inference time, the deterministic generator used the input image, along with stochastically sampled latent codes, to produce randomly sampled outputs. A typical problem in previous methods is the issue of mode collapse where only a small number of real samples get represented in the output. This was overcome in the BicycleGAN, by combining the ideas of the Conditional Variational Autoencoder GAN [74] and Conditional Latent Regressor GAN [75, 76] into a hybrid model. In cVAE-GAN the latent encoding was learned from real data, but at test time, the randomly sampled latent code may not have yielded realistic images as the discriminator did not have opportunity to learn (and thus critique the generator) from the results sampled from the prior during training. In cLR-GAN, the latent space was easily sampled from a simple distribution, but the generator was trained without the benefit of seeing ground truth input-output pairs. Hence the BicycleGAN was trained to enforce the connection between latent encoding and output in both directions jointly.

2.3 Reinforcement learning

While supervised learning can be used to address problems such as classifying images or translating texts however may be unsuitable in scenarios such as learning to play the game Go. If supervised learning was to be used, one could gather a dataset where the inputs of the model are all the possible game states and the output labels being the *optimal* move for that particular state. However, creating this dataset would be expensive and unfeasible as the number of possible states is large (2.082×10^{170}). Furthermore, such an approach relies on imitating a human expert which may not provide the optimal strategy.

The Reinforcement Learning (RL) framework attempts to find the best action for a given state through trial and error. The RL agent learns the optimal strategy by sampling actions and then observing which one leads to the desired outcome. When compared to supervised learning, the agent learns this optimal action not from a label but from a time-delayed label called a *reward*. The reward is a scalar value describing whether, after performing a sequence of actions, the agent had reached its goal (with varying degrees) or not. In other words, the agent is told what the best outcome should be through the reward. It is not given instructions as to *how* to achieve this reward but instead must discover the strategy through trial and error.

More formally, the RL problem is formulated as a Markov Decision Process (MDP) with an agent that makes decisions in a stochastic environment, in order to achieve a goal. The decision process is formulated under the Markov property which states that in a stochastic environment, the conditional probability distribution of future states of the agent (conditional on both past and present states) depends only upon the present state and not on the sequence of events that preceded it [77].

The MDP in this context, is defined as a tuple that contains:

1) a state space (S)- a set of states the agents can be at a given time, t. s_t denotes the state of the agent at t-th time step,

2) an action space (A)- the set of actions the agent can perform at a given time, where a_t is the action of the agent at time step t,

3) a reward function ($R(s_t, a_t)$)- the reward given to the agent by the environment by performing action a_t at state s_t,

4) a transition model ($T(s_t, a_t, s_{t+1})$)- this specifies the dynamics of the environment which could be given to the agent or learned. It denotes the probability

of going to state s_{t+1} by performing action a_t in state s_t.

Before delving into different algorithms for reinforcement learning, the following definitions will be used:

- *policy*, π - which defines the learning agent's way of behaving in a certain state, s_t. It is a mapping from the perceived states of the environment to the actions being taken when in those states,
- *value function*, $V_\pi(s)$- the expected reward when starting in the state s and following π thereafter,
- *Q-value*, $Q_\pi(s, a)$- the expected reward when starting from s and taking action a, and then thereafter following policy π.

2.3.1 Temporal difference learning

Temporal Difference (TD) learning [78] is a class of reinforcement learning methods that can be used to estimate the aforementioned value functions. These algorithms do not require the transition dynamics of the environment and hence is used in *model-free* reinforcement learning. These methods sample from the environment the resultant states and rewards for chosen actions and performs updates on the current values of states based on their current estimates. In particular, it estimates the current estimate of the value function by *bootstrapping* samples.

If the value functions were to be calculated without estimation, the agent would need to wait until the final reward was received before any state-action pair values can be updated. Once the final reward was received, the path taken to reach the final state would need to be traced back and each value updated accordingly, according to the equation

$$V(s_t) \longleftarrow V(s_t) + \alpha(R_t - V(s_t)), \tag{2.31}$$

where s_t is the state visited at time t, R_t is the reward at time t and α is a constant parameter known as the learning rate which determines how much $V(s_t)$ is updated.

On the other hand, with TD methods, an estimate of the final reward is calculated at each state and the state-action value updated for every step of the way instead of waiting until the end of the episode.

The algorithm starts by initializing a table $V(s)$ arbitrarily, a value for each state in the MDP. Then, the policy π is evaluated, obtain a reward r_t for each timestep and the value function is updated according to

$$V_(s_t) \longleftarrow V(s_t) + \alpha\big[R_{t+1} + \gamma V(s_{t+1}) - V(s_t)\big]. \tag{2.32}$$

The value functions are updated using the rewards gained from executing actions determined by some policy. The strategies that are used for selecting an action which aim to balance the trade-off between *exploitation*, where the agent uses the current estimates of the action-value function to maximise its expected reward, and *exploration*, where agent chooses less explored actions to obtain a better estimate of the value function. There are three common strategies:

- *greedy*- the action with the highest estimated expected reward, called the greediest action, is chosen. This strategy maximises exploitation but has little exploration and therefore may not discover the optimal policy.

- ϵ-*greedy*- the greediest action most of the time with some probability ϵ (say 0.9) and for some probability $1-\epsilon$, another action that is not the greediest, is selected uniformly at random independent of the action-value estimates. This method ensures that with enough trials, each actions will be tried infinite number of times, thus ensuring optimal actions are discovered.

- *softmax*- the probability of a chosen action is weighted according to their action-value function. In the case of ϵ-greedy where non-greediest actions are chosen uniformly at random, this means that the worst possible action is just as likely to be chosen as the second best. The softmax strategy ensures the worse actions are less likely to be chosen and could be a strong approach when the worst actions are unfavourable.

TD methods can be learned in two different manners, *on-policy* and *off-policy* learning. On-Policy TD methods learn the value of the policy that is used to make decisions. The value functions are updated using results from executing actions determined by some policy such as ϵ-greedy or softmax. It attempts to evaluate or improve the policy that is used to make decisions. Conversely, off-policy methods learn different policies for behaviour and estimation. It updates the estimated value functions using actions which have not actually been tried. This is in contrast to on-policy methods which update value functions based strictly on experience. This means that during execution, the agent can learn behaviours that it did not exhibit during the learning phase.

2.3.2 Q-learning

Q-learning [79] is an off-policy model-free approach to finding the optimal policy, π^*, which outputs the best action a_t for a given state, s_t. The Q-learning algorithm goes as follows. First the Q-values ($Q(s,a)$), is initialised for each state-action combination. Then for each episode, for every step in the episode, the current state, s_t is observed, and an action is chosen, a_t based on a random action selection policy (e.g. ϵ-greedy). The resulting reward, r_t and state, s_{t+1} is observed and then the Q-value for the state-action pair is updated using,

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\big[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\big]. \tag{2.33}$$

The reason that Q-learning is off-policy is that it updates its Q-values using the Q-value of the next state s_{t+1} and the greedy action a'. In other words, it estimates the return (total discounted future reward) for state-action pairs assuming a greedy policy was followed despite the fact that it is not originally following a greedy policy [77].

Algorithm 1: Q-learning algroithm

Input: Number of episodes, N

Output: Q-values $Q(s, a)$

Initialise $Q(s, a) = 0, \forall s \in S, a \in A$

for *i=1,2,...,N* **do**

    repeat

        Choose a_t using policy derived from $Q(s_t, a_t)$ (e.g. ϵ-greedy)

        Execute a_t and observe r_t, s_{t+1}

        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\big[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\big]$

        t= t+1

    until s_t *is terminal*;

end

return *Q(s,a)*

2.4 The Brain and Mind

2.4.1 Mental Illness

Mental illness are disorders that affect a person's mood, thinking and behaviour. It covers a spectrum of disorders that vary in how severe they are and how long they last. In some cases, symptoms can be managed with a combination of medications and psychotherapy. While their causes are still an open research question, they are thought to be caused by a variety of genetic and environmental factors such as [80]

1) inherited genes that can increase the risk of developing mental illness,

2) environmental exposure to toxins or alcohol before birth have been linked to mental disorders,

3) the impairment of neurotransmitters which can change the nerve receptors and nerve systems.

. Examples of mental illness include depression, bipolar disorder and schizophrenia. The diagnosis for these disorders is largely based on the Diagnostic and Statistical Manual of Mental Disorders [81] where it is a handbook for the, descriptions, symptoms and classification of mental disorders using a common language and standard criteria.

Depression

Depression has been recognised as a clinical syndrome for over 2000 years, where a number of ancient writers described the disease under the classification of melancholia. Hippocrates, in 4th century B.C.E, referred to the diseases swings similar to mania and depression, Aretaeus, a physician in the second century C.E described patients with depression as "sad, dismayed, sleepless" and eventually who complained of the futility of life and contemplated suicide. In modern textbooks, depression is defined in terms of the following attributes:

1) A specific alteration in mood or changes in the person's feelings such as a dejected mood.

2) Negative feelings towards self, and a negative outlook towards life.

3) Motivational manifestations such as the loss of positive motivation, or wanting to avoid or escape their usual routines or having suicidal wishes.

4) Vegetative and physical manifestations such as the loss of apetite, sleep disturbance and loss of libido,

5) Delusions such as the sense of worthlessness [82].

Bipolar Disorder

Bipolar disorder is a chronic disorder that is attributed with fluctuations in mood state and energy which affects more than 1% of the world's population. It is characterised by bouts of mania which in many ways is opposite of depression. During a manic episode, individual's experience elevated mood, over-activity with a lack of sleep and an increased optimism that impairs the individual's judgement. However, a depression may alternate with manic episode. These large changes in mood is one of the major causes of disability among young people. Furthermore patients are at very high risk of death by suicide, up to 20 times higher than the general population [83, 84]. Unlike unipolar depression, bipolar disorder usually has an earlier age of onset, more frequent episodes of short duration, has an abrupt onset and offset and is triggered by stressors at early stages [81].

The discovery that lithium had a considerable affect in mood stabilisation, which suggested that the classification and mechanistic causes of bipolar disorder could be explained by a biologic pathophysiological framework. In particular, these mood disorders were though to result from an imbalance of monoaminergic neurotransmitter systems such as the dopamingeric neurotransmitter systems. However, no singular dysfunctions of these neurotransmitter systems has been identified. That being said, bipolar disorder is one of the most inheritable psychiatric disorders where about 50 percent of patients with bipolar illness have a family history of the disorder [83].

Schizophrenia

Schizophrenia is a syndrome that is predominately defined by observed signs of psychosis. It is presented with paranoid delusions and auditory hallucinations late in adolescence or in early adulthood between the ages 18-25. These are sometimes accompanied with social withdrawal, decreased emotional expression, and apathy. Furthermore, those with schizophrenia experience social problems such as long-term unemployment, poverty, homelessness, a higher suicide rate and a lower life ex-

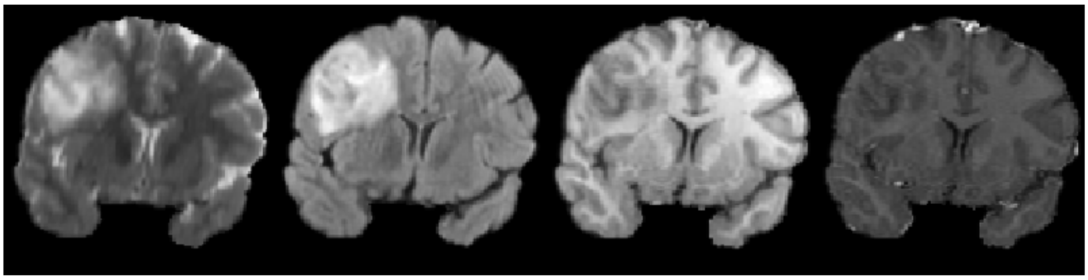


Figure 2.5: Examples of the different modalities of MRI scans of a coronal slice of a low grade glioma (brain tumour) in the BraTS dataset brain using different MRI sequences. From left to right: T2, FLAIR, T1 and T1c.

pectancy of 20 years [85].

The understanding of schizophrenia has changed over the last two centuries. In the early twentieth century emphasised the mind, where schizophrenia was characterised by the fundamental disorder of thought and feeling being disturbances of associations, affect, ambivalence and autistic isolation. It was seen as a psychotic reaction which resulted in a fragmented ego caused by environmental experiences and trauma. Later in the second half of the twentieth century, schizophrenia was seen as a "dopamine disorder" which resulted in the development of neuroleptic medications. Whilst they reliably reduced delusions and hallucinations, they did not improve the functional well-being of patients, as many are still considered disabled. Another framework for understanding the causes for schizophrenia is genetics as the illness is highly inheritable [86].

2.4.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a noninvasive imaging technique that uses strong magnets and low-energy radio-frequency signals to record the response from certain atomic nuclei within the body. In particular, when analysing tissue, the hydrogen nuclei is mostly used [87].

Without the presence of a magnetic field, the magnetic moments of nuclei are distributed at random and thus the net magnetisation factor is zero. However, under a strong external magnetic field, the spin of nuclei is aligned parallel or antiparallel to the external field with a net magnetisation vector parallel to the external magnetic field. Furthermore, the individual nuclei do not directly line up with the magnetic field but precess around the direction of the external field. The frequency of this precession is given by Larmor equation:

$$F = \frac{\gamma B_0}{2\pi}, \tag{2.34}$$

where F is the precessional frequency or the Larmor frequency, B_0 is the strength of the magnetic field and γ is the gyro magnetic ratio or the ratio of the magnetic moment to the angular momentum of the hydrogen nucleus.

When the nuclei is under the presence of the external magnetic field, a measurable signal is created by applying a radio frequency (RF) pulse at the Larmor frequency (resonant frequency), which flips the net magnetisation of the nuclei spines to 90°. This rotating net magnetisation vector induces an AC in a receiver coil which is placed around the patient [87].

When the RF frequency is stopped, the magnetisation vector returns to its previous equilibrium state under the external magnetic field. The time required to return to equilibrium is known as the *relaxation time*. The process of realignment to the external magnetic field is called the longitudinal relaxation process. This process is characterised by the T1-relaxation time which is defined as the time required for the system to recover to 63% of its equilibrium value after being exposed to the RF pulse. Different human tissue have different T1 values.

There is also another relaxation process, now in the traverse direction which is independent to the longitudinal direction. This second process of relaxation is due to the spins precessing around the magnetisation vector and the small differences cause by local magnetic inhomgeneities. This relaxation time is described by the T2 relaxation time and is different to the T1 values for various tissue [87].

The MRI also has the potential to visualize the difference in T1 and T2 of different tissue to produce different contrasts between soft tissue. By manipulating scanning variables such as the time of repetition (TR), which is the time for the next repetition of RF pulses, and the time of echo (TE), which is the interval between a second RF pulse and an echo signal, the MRI is able to produce contrasts dependent on T2 differences or T1 differences or neither (meaning it will only depend on the proton density of the tissue) [88].

A conventional MRI sequence is the *spin-echo* pulse sequence where it is comprised of a repeated sequence of two RF pulses, one called the 90° pulse and the other 180° pulse. Initially, the magnetic moment of a group of spins, such as protons is in its equilibrium position. Then a RF pulse is released which causes the atoms' magnetisation to undergo a 90° displacement to the transverse plane. The tissues show a distribution of frequency of precession due to local magnetic field inhomogeneities. As the net moment precesses, some spins slow down due to lower local field strength (and so begin to progressively trail behind) while some speed up due to higher field strength and start getting ahead of the others. This loss can be reversed by applying a 180° pulse, whereby the fast moments catch up with the main moment and the slow moments drift back toward the main moment. Once the moments have rephased, the *echo* can be sampled [89].[1]

For the *spin-echo* pulse sequence, the TR is defined by the time between the repetition of 90° pulses and the TE is the duration between the middle of a 90° pulse and the middle of an echo. By adjusting these parameters and others like the gradients, MRI machines are able to create sets of images with a particular appearance, called MRI modalities. These modalities include T1-weighted (T1), T2-weighted (T2), T1 with gadolinium enhancing contrast (T1c), Fluid Attenuated Inversion Recovery (FLAIR) and Diffusion Weighted Imaging (DWI).

[1]See `https://en.wikipedia.org/wiki/Spin_echo` for an animation of a spin echo sequence.

T1-weighted

T1-weighted images are produced by using short TE and TR times. The contrast and brightness of the image are predominately determined by longitudinal relaxation time (T1) properties of tissue.

For instance, fat tissue quickly realigns its longitudinal magnetisation with the main magnetic field and therefore appears bright on a T1-weighted image. Conversely, water has much slower longitudinal magnetisation realignment after an RF pulse and therefore has low signal. This causes fluid such as urine or CSF to appear dark on the image.

If sufficiently short TR times were not used during the MRI sequence, then the protons would recover their alignment with the main magnetic field and hence the image would be uniformly intense. By having TR times that are shorter than the different tissues' recovering time allows for the different contrast of the tissue [90].

Some brain tissue can also be distinguished using this particular MRI sequence, such as white matter and grey matter where this contrast is driven by the differences in their fat content. White matter is primarily comprised of axons that carry nerve impulses between neurons. Axons are insulated by myelin which is a substance that is fat (lipid) rich. Due to this fat content, it appears white on a T1 image. On the other hand, grey matter is mostly composed of neuron cell bodies and non-neuron brain cells called glial cells. As these cells are not surrounded by myelin, have an intermediate signal intensity and thus appears grey on T1 image [91].

T1 with gadolinium enhancing contrast

T1-weighted imaging can also be performed while infusing gadolinium creating a T1 with gadolinium enhancing contrast image. Gadolinium is a non-toxic paramagnetic contrast enhancement agent. When injected during the scan, Gadolinium shortens the T1 time thus appears very bright on T1-weighted images. T1c images are especially useful in looking at vascular structures and breakdown in the blood-brain barrier like tumours, and diseases that cause an inflammatory response like multiple sclerosis [92].

T2-weighted

T2-weighted imaging relies upon the transverse relaxation time of protons and tends to use long TE and long TR MRI sequences. Each tissue has an inherent T2 value, but external factors (such as magnetic field inhomogeneity) can decrease the T2 relaxation time. A refocusing pulse is used on spin-echo sequences helps to mitigate these extraneous influences on the T2 relaxation time [91].

T2 imaging creates large signal for fluid in the image (as opposed to dark in T1 images) and fat tissue tends to appear an intermediate-bright on the image. When imaging the brain, white matter generally appears darker than grey matter using a T2 image.

Fluid Attenuated Inversion Recovery

Fluid Attenuated Inversion Recovery is a particular MRI sequence that uses a particular pulse sequence called inversion recovery (IR) to remove the signal of CSF in the resulting image. The IR pulse sequence is a spin echo pulse sequence that is also preceded by a 180° RF pulse.

The purpose of inverting pulse is to flip the initial longitudinal magnetisation to the direction opposite of the main magnetic field. These tissues with inverted magnetisation moments undergo T1 relaxation as they variably seek to re-establish magnetisation along the longitudinal direction. When spin echo signal generation begins (at the 90°-pulse), the initial longitudinal magnetisations of different tissues are now separated based on their different intrinsic T1 relaxation times. More specifically, the spin echo 90° readout pulse is applied at the exact time when longitudinal magnetisation reaches the null point for CSF and hence ideally removes the signal generated by CSF. The time between the inverting pulse and the spin-echo pulse is called the time to inversion (TI) [89].

2.4.3 Computational Decision Making

A computational model of decision-making uses past experiences such as chosen actions and value of rewards and outputs predictions about future actions. These models are learned using the empirical data gathered from the behaviour of human participants and if these models are fitted different groups of participants (e.g. healthy vs bipolar), the differences in the learned parameters can get insight about the behavioural differences of these groups.

Several computational learning tasks, such as the bandit task and the two-stage task, have been used to analyse the behaviour of different groups of participants.

Bandit Task

The bandit task [77] comes from imagining a gambler (agent) at a row of slot machines (sometimes known as "one-armed bandits"), who has to decide which machines to play, how many times to play each machine and in which order to play them, and whether to continue with the current machine or try a different machine. The rewards from each machine is probabilistic, each with its own probability distribution. The objective of the agent is to maximise the sum of rewards earned through pulling a sequence of levers. Most importantly, the gambler faces a trade-off between exploitation, where the agent leverages their current information to maximise their reward by choosing the level with the highest expected pay out, and exploration to get more information about the expected payoffs of the other machines.

More formally, the multi-armed bandit is a set of real distributions, $B = \{R_1, .., R_K\}$, with each distribution being associated with the rewards delivered by one of the K levers. Each distribution is associated with a mean value of reward however which may not necessarily remain static over iterative plays. The agent iteratively plays one lever per round and observed an associated reward. Their task is to maximise

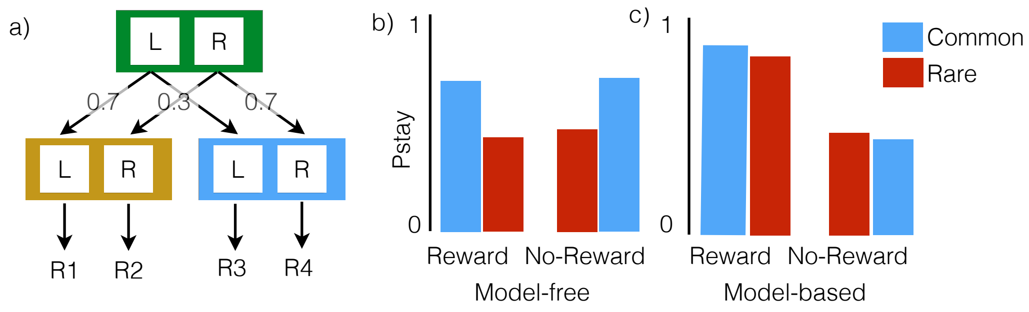


Figure 2.6: **a)** Diagram of the two-stage task showing the states, actions and the transition probabilities of each state. For example, at stage 1 (green state) if "L"" was chosen, 70% of the time, the stage 2 state will be gold state (the common transition). However choosing "L" at stage 1 could transition to the blue state (rare transition) 30% of the time. **b)** The probability of the RL agent (**left**: model-based, **right**: model-free) staying on the same action at stage 1 and stage 2 (e.g. choosing "L"" in both stages) depending whether a rare or common transition was observed in the current episode and if a reward was received in the previous episode.

the sum of collected reward over H number of rounds.

This bandit task has been used for example, to study the effects of age and dopamine availability on striatal and prefrontal mechanisms [93] as well as differences between patients with unipolar or bipolar depression and healthy participants [18].

Two-stage Task

The two-stage task has been used to demonstrate the distinct systems for decision making, namely the habitual and goal-directed actions systems. The habitual decision making system has a principle that an action followed by reinforcement is more likely to be repeated in the future. This mechanism is akin to TD learning where the learning is "model-free" which ignores the structure of the task but instead learns directly by the reinforcement of successful actions. On the other hand, the goal-directed learn is similar to a "model-based" RL agent which uses the given model of the task to evaluate candidate actions [94].

The two-stage task is described as follows. On each trial of the task the agent made two binary decisions at stage 1 and stage 2. The initial choice at stage 1 led probabilistically to either of two, stage 2 "states". In turn, these stage 2 states both demanded another choice of two options, each of which was associated with a different chance of delivering a binary reward (1 or 0). The choice of a stage 1 action led predominantly (70% of the time) to an associated one of the two stage 2 states, which are referred to as "common" transitions, and this relationship was fixed throughout the experiment. Conversely, 30% of the time, the stage 1 action led to the alternate stage 2 states, labeled as the "rare" transition.

The task consists of three states (first stage: s_A; second stage: s_B and s_C), each with two actions (a_A and a_B). The goal of the model-based and model-free agents is to learn a state-action value function $Q(s, a)$. On trial t, we donate the first-stage state by $s_{1,t}$, the second-stage state by $s_{2,t}$, the actions in each stage as $a_{1,t}$ and $a_{2,t}$ respectively and the rewards for the first-stage as $r_{1,t}$ (which is always 0) and second-stage, $r_{2,t}$.

The model-free algorithm was SARSA(λ) TD learning [95]. At each stage i of each trial t, the value of the visited state-action pair was updated according to

$$Q_{TD}(s_{i,t}, a_{i,t}) = Q_{TD}(s_{i,t}, a_{i,t}) + \alpha(r_{i,t} + Q_{TD}(s_{i+1,t}, a_{i+1,t}) - Q_{TD}(s_{i,t}, a_{i,t})), \quad (2.35)$$

where α is the learning rate. We set α to be the same for the first- and second-stage in simulations. We define $Q_{TD}(s_{3,t}, a_{3,t}) = 0$ since there is no further value in the trial apart from the immediate reward $r_{2,t}$.

Model-based reinforcement learning refers to learning optimal behavior indirectly by learning a model of the environment by taking actions and observing the outcomes that include the next state and the immediate reward. The agent attempts to learn optimal policies by predicting the outcomes of actions when interacting with the environment. In accordance to the structure of the two-stage task, this amounts to agents deciding which first-stage actions maps to which second-stage and second, learning immediate reward values for each of the second-stage options.

Daw et al. [94] characterized transition learning by assuming subjects simply chose between the two possibilities $P(s_B|s_A, a_A) = 0.7$, $P(s_C|s_A, a_B) = 0.7$, $P(s_C|s_A, a_A) = 1 - P(s_B|s_A, a_A) = 0.3$ and $P(s_A|s_A, a_B) = 1 - P(s_C|s_A, a_B) = 0.3$.

At the second-stage (the only one where immediate rewards were offered), the problem of learning immediate rewards is equivalent to that for TD above, since $Q_{TD}(s_{t,2}, a_{2,t})$ is just an estimate of the immediate reward, $r_{2,t}$. With no further stages to anticipate, the model-free and model-based approaches coincide at the second-stage, and we define $Q_{MB} = Q_{TD}$ for states s_B and s_C.

Next, using Bellman's equation, the model-based values of the first-stage actions is defined as

$$Q_{MB}(s_A, a_j) = P(s_B|s_A, a_j) \max_{a \in \{a_A, a_B\}} Q_{TD}(s_B, a) + P(s_C|s_A, a_j) \max_{a \in \{a_A, a_B\}} Q_{TD}(s_C, a),$$

which are recomputed at the end of each trial using the current transition probabilities and rewards.

The state-action values are connected to choices by using a softmax choice rule, which assigns a probability to each action according to

$$P(a_{i,t}|s_{i,t}) = \frac{\exp(\beta Q(s_{i,t}, a))}{\sum_{a'} \exp(\beta Q(s_{i,t}, a'))}, \tag{2.36}$$

where β is the inverse temperature parameter which determines the propensity to explore, equivalently controlling the impact of receiving a reward from an action on repeating that action in future trials.

By applying these two learning algorithms, model-free and model based, to the two-stage task, different behaviours are observed as shown in Figure 2.6. The model-free algorithm predicts that a stage 1 choice resulting in a reward is more likely to be repeated on the subsequent trial ($P(stay)$), regardless of whether that reward occurred after a common or rare transition. On the other hand, a model-based learning algorithm predicts that a rare transition should affect the value of the other stage 1 option, leading to an interaction between whether a reward was received and the type of transition observed [94].

The two-stage task has been used to demonstrate the deficits in goal-directed control were more strongly associated with symptoms of compulsive behaviour and intrusive thought [96] and has been used to analyse the process of habit formation in healthy individuals [97].

2.5 Summary

So far, an overview of deep learning methods, which are typically used in supervised learning problems, ranging from feed-forward neural networks through to recurrent neural networks were discussed and the methods for training these models. Furthermore, we discussed other types of machine learning problems, unsupervised learning where the goal is to learn patterns in the data without any preexisting labels, and reinforcement learning that attempts to find the best action for a given situation through the process of exploration and exploitation. We completed the background with descriptions of mental illness and various tools, MRI and computational decision tasks, that are used to analyse the brain and mind.

The next sections will introduce machine learning methods to several tasks to improve current tools in neuroscience. We propose the use of the CycleGAN for unsupervised normalisation of two distinct MR images from different sites in order to improve existing methods that pool data in order to increase sensitivity and statistical power. The thesis further extends the model to be used in semi-supervised learning to translate different modalities of MR images of patients with brain tumours and lesions. Lastly, a deep learning architecture is presented that models human behaviour on sequential tasks whilst also explaining the characteristics that defines the differences between different groups of participants.

Chapter 3

Correction of MRI Multisite Differences

3.1 Introduction

One of the biggest challenges in the translation of neuroimaging findings into clinical practice is the need to validate models across large independent samples and across data obtained from different MRI scanners and sites. Combining multiple samples increases the overall sample size, overcoming a limitation common to many neuroimaging studies. However it also introduces heterogeneity into the sample from differences in scanner manufacturer, MRI protocol, variation in site thermal and power stability, as well as site differences in gradient linearity, centring and eddy currents. Therefore, images from different sites have the potential to introduce bias that can either mimic or obscure true changes or even worse, produce results that could be driven by the artifactual site differences. This can make the interpretation, reliability and reproducibility of findings difficult. Despite these issues, pooling data provides the opportunity to address a major source of concern regarding the low statistical power of published studies, especially when larger studies are not feasible due to financial constraints or recruitment is difficult because a particular disorder is rare at a specific geographical location [30].

Given the considerable incentives to pool data, there is a relative paucity of methods available to correct for site-specific differences in MR images. The majority of approaches are usually applied during data acquisition, for instance, using a common phantom across sites to calibrate and reduce differences in field homogeneities. However, these *a priori* methods require careful planning and are not applicable to data sets that have already been collected or other *post hoc* forms of data pooling. Site differences can also be addressed in a *post hoc* fashion by treating the site as a covariate in the analysis for evaluation of confounding effects. However, the interaction between the usually unknown site-specific effects and the true brain effects on the MRI signal seem to be highly complex and nonlinear such that the inclusion of the covariate can also introduce bias [98].

3.2 Related Work

In the context of predictive modelling in neuroimaging, previous works have focused on including variables such as age, gender in predictive modelling [99, 100]. Such variables are highly correlated with imaging data but are not relevant for clinical analysis and are known as *confounds*. The standard approach is to remove the contributions of these variables through regression based methods [100]. As such, image data is adjusted such that the adjusted data can be considered as being produced by subjects that have identical confounders and then this adjusted data is then used to estimate the predictive model. These methods typically fit a linear model between each image feature in turn and the confounders as features [98].

A second approach is to explicitly include the confounder as predictors in the predictive model and are treated similarly to the image features [101]. The advantage of this approach is that it allows the model training procedure to produce a model that predicts well, whether or not there is bias in the training sample. However the may fail due to problems of *covariate shift* where the distribution of the predictor features in the training sample does not match the distribution in the population of interest [98, 102]. Particularly when confounders are present, the relationship between the confounders and the image data will cause the training data be unrepresentative of the population of interest [98].

Recent advances in computer vision due to the application of artificial neural networks suggests there may be a novel *post hoc* solution to remove non-linear bias in MR images. For example, superior performance in non-linear, multivariate pattern classification problems such as Alzheimer's disease classification, brain lesion segmentation, skull stripping and brain age prediction have been achieved using deep learning networks [1–5]. Deep learning provides some unique advantages for high-dimensional data such as MRI data, since it does not require extensive feature engineering. Furthermore, deep learning has produced important advances in generative modelling. Generative modelling involves learning to estimate a given distribution in order to produce examples from that distribution. For example, after being trained on a set of images, the model is able to generate a new, 'unseen' sample from the training set. Generative modelling is considered a much more difficult task than pattern classification, as the output of these models are typically high dimensional and a single input may correspond to many correct answers (e.g. there are many ways of producing an image of a cat).

One class of generative models, known as generative adversarial networks succeed through the idea of adversarial training, where the model's training process can be described as a game between two players. One player is called the generator where it attempts to create samples from the same distribution as the observed data. The other player is the discriminator where its function is to examine the fake samples from the generator and real samples from the observed data and to classify the generated and observed samples as either real or fake (see Section 2.2.3 for more details.

Here, we propose an algorithm that uses GANs to transform a set of images from a given MRI site into images with characteristics of a different MRI site. Its purpose is to correct for differences in site artefacts without the need for *a priori* calibration using phantoms or significant coordination of acquisition parameters. This algorithm can be treated as a 'black box' without knowledge of the artefacts present in the dataset and can be applied *post hoc* after acquisition to two or more unpaired sets of imaging data. Importantly, as we demonstrate, the correction occurs without any apparent loss of information related to gender or clinical diagnosis.

3.3 Unsupervised Domain Adaptation for Neuroimaging

3.3.1 Participants

Structural (T1-weighted) MR brain images were obtained (N = 313) from preexisting MRI studies conducted at two different sites (site A and site B). The cohort from each site contained two diagnostic groups (schizophrenia and healthy adults), however these groups were not evenly distributed over sites (see Table 3.1). All clinical cases met DSM-IV criteria for their disorder with no other Axis I disorders, on the basis of either the Mini-International Neuropsychiatric Interview [103] or the Structured Clinical Interview for DSM-IV Axis I and II Disorders [104]. Participants were aged 18-65 years and spoke fluent English. Exclusion criteria included the presence of an organic brain disorder, brain injury with post-traumatic amnesia, mental retardation (WAIS-III IQ score less than 80), movement disorders and recent (within 6 months) substance dependence or electroconvulsive therapy. Healthy adults were also screened for the absence of personal or family history of any DSM-IV Axis I disorder.[1]

[1]This research was conducted under approval from the University of Sydney Human Research Ethics Committee, HREC 2014/557.

Table 3.1: Subject and gender distribution across sites (m:male, f:female)

	Site A	**Site B**	**Total**
Control			
n	41	101	142
age ± SD	29.7±13.1	31.2±8.7	31.2±10.1
m/f	23/18	52/49	75/67
Schizophrenia			
n	17	154	171
age ± SD	44.8±11.1	38.0±9.5	38.7±9.8
m/f	7/10	57/97	64/107
Total			
n	58	255	313
age ± SD	34.1±14.1	35.3±9.7	35.1±10.7
m/f	30/28	109/158	139/174

3.3.2 MR Scanner, image data and preprocessing

Data were collected from two different MRI sites: Site A hosted a Phillips Achieva 3T with a 8-channel head coil and receiver (NeuRA, Randwick NSW, Australia); and Site B hosted a GE Discovery MR750 3T with a 8-channel head coil and receiver (Brain and Mind Centre, Camperdown NSW, Australia). T1-weighted image volumes were acquired using a standard but scanner-specific MPRAGE acquisition sequence. T1 images from Site A were acquired with a 3D Fast Spoiled Gradient Recall Echo (FSPGR) sequence with SENSE acceleration; 8.3-ms TR, 3.2-ms TE; and 11 degree flip angle, and comprised of 180 sagittal 1-mm slices in a 256 x 256 matrix (1 mm isotropic voxel dimensions). Images from Site B were acquired with a 3D Turbo Field Echo sequence (TFE) with ASSET acceleration; 7.192-ms TR, 2.732-ms TE; and 12 degree flip angle, and comprised of 176 sagittal 1-mm slices in a 256 x 256 matrix (1 mm isotropic voxel dimensions).

Image preprocessing was designed to remove as much of the site differences as possible given standard tools available, before applying the novel GAN method described in the next section. All preprocessing occurred in SPM12, running under Matlab 8.4 (Math Works, Natick, MA, USA). After checking for scanner artefacts and gross anatomical abnormalities for each image, we reoriented the original images along the Anterior-posterior commissure (AC-PC) line and set the AC as the origin of the spatial coordinates to assist the normalisation algorithm. The unified segmentation procedure in SPM12 was used to segment all the images into mean corrected grey matter (GM), white matter (WM), cerebrospinal fluid (CSF) space, i.e. maps of probability values representing the probability of a voxel containing a specific tissue type. Mean correction was applied to remove site differences in the bias field. A fast diffomorphic image registration algorithm [105] was used to warp the GM partitions into a new study-specific reference space representing an average of all 313 subjects included in the analysis. As an initial step, a set of study-specific templates and the corresponding deformation fields, required to warp the data from each subject to the new reference space, were created using the GM partitions [106]. Each subject-specific deformation field was used to warp the corresponding GM partition into the new reference space with the aim of maximising accuracy and sensitivity [107]; the warped GM partitions were affine transformed into the MNI space and an additional 'modulation' step was used to scale the GM probability values by the Jacobian determinants of the deformations in order to ensure that the total amount of GM in each voxel was conserved after the registration [108–110]. After this preprocessing, we obtained bias-field corrected, modulated, normalised GM density maps from which we divided each brain volume into 2D sagittal slices to be used to train the GAN model described below.

3.3.3 Generative Adversarial Networks

Rather than removing any remaining scanner artefacts and biases from the images, we seek to transform one set of images from a site to images that come from the distribution of images from the other site, while still preserving the important features of the original images.

Notation: In the following, we have defined capital bold font, $\mathbf{X}$, as a matrix or a set of images and lower case bold font, $\mathbf{x}$, as a vector or one example image. G_θ, D_ϕ

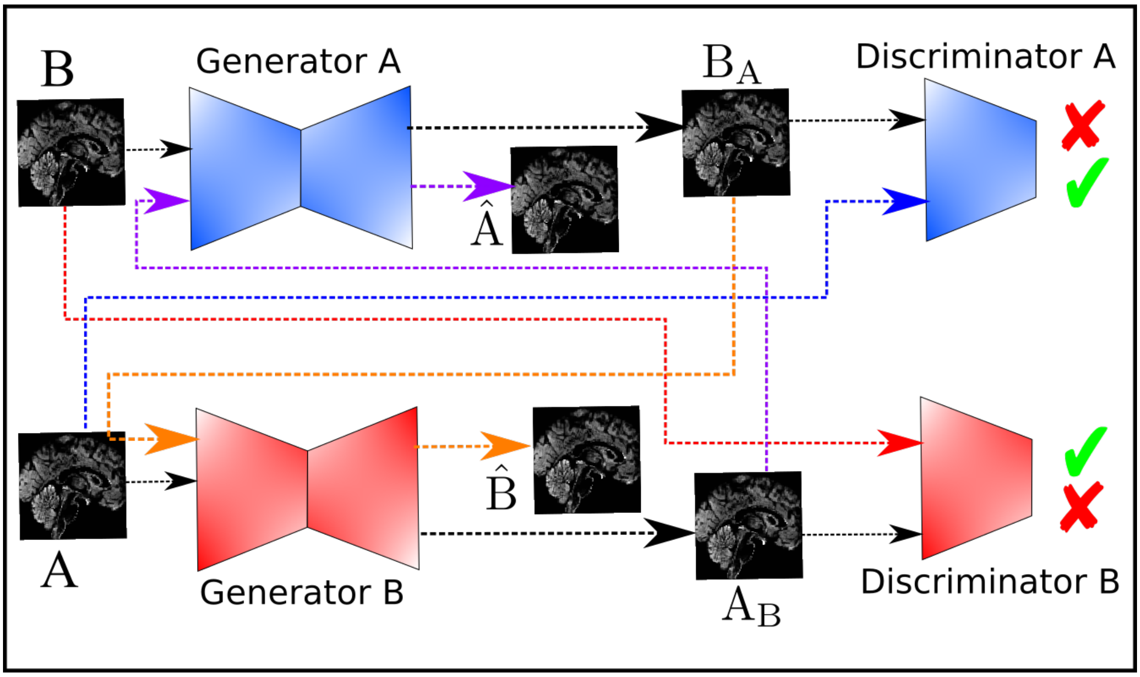


Figure 3.1: Architecture of the CycleGAN.

denotes a mapping function parameterised by θ and ϕ, respectively. $P(\mathbf{X})$ indicates the probability distribution for the imaging set $\mathbf{X}$, and $\hat{P}(\mathbf{X})$ is an estimate of that probability distribution.

The problem at hand can be described as image-to-image translation in the computer vision literature where the goal is to learn a mapping function between a set of MRI images from domain $\mathbf{X}$ and another set of images from domain $\mathbf{Y}$; learn $G : \mathbf{X} \to \mathbf{Y}$ such that $G(\mathbf{x})$ for each $\mathbf{x} \in \mathbf{X}$ is indistinguishable from the set of images from domain $\mathbf{Y}$.

The CycleGAN [35] and DiscoGAN [111] have been developed to learn cross domain relationships between sets of natural objects such as from horses to zebras, edges to photos and Monet artworks to realistic photos. The advantage of these models is that they do not require paired sets of training samples, $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$, which is often difficult to obtain for neuroimaging data, and instead only require unpaired imaging data consisting of a source set $\{\mathbf{x}_i\}_{i=1}^{N} \in \mathbf{X}$ and target set $\{\mathbf{y}_j\}_{j=1}^{M} \in \mathbf{Y}$, without any $\mathbf{x}_i$'s necessarily corresponding to any $\mathbf{y}_j$'s. These models attempt to transform the underlying distribution of $P(\mathbf{X})$ to an estimate of $P(\mathbf{Y})$, $\hat{P}(\mathbf{Y})$, through G while still preserving the important features of the original sample, $\mathbf{x}_i$, but also merging these with the particular characteristics of $P(\mathbf{Y})$.

To learn this mapping function, an adversarial training regime was utilised using the GAN formulation. The generator, G_θ, represented as a convolutional neural network defined by parameters θ, takes as input, images from $\mathbf{X}$ and transforms these images, $G_\theta(\mathbf{x})$, as if they were sampled from $P(\mathbf{Y})$. The discriminator, D_ϕ on the other hand, is a supervised classifier represented as a convolutional neural network. The discriminator observes two inputs, the observed images from $\mathbf{Y}$ and generated samples $G_\theta(\mathbf{x})$. The goal of the discriminator is to output a probability that its inputs are either real or fake, with the true labels being observed samples as real and generated samples as fake. The discriminator attempts to learn that its output from samples of $\mathbf{Y}$, $D_\phi(\mathbf{y})$, are given to be given values near 1 and inputs from the generator, $D_\phi(G_\theta(\mathbf{x}))$, to be values close to 0. However, at the same time, the generator will attempt to make the quantity, $D_\phi(G_\theta(\mathbf{x}))$ to approach 1. At equilibrium, $D_\phi(\mathbf{y}) = \frac{1}{2}$ for all $\mathbf{y}$ and $G_\theta(\mathbf{x})$ which means that the discriminator is unable to distinguish between real and generated samples [63].

More specifically, the Least Squares GAN (LSGAN) [112] is used to train the discriminator and generator, where the discriminator's objective function is

$$\min_\phi \frac{1}{2} E_{\mathbf{y}\sim p(\mathbf{Y})}[(D_\phi(\mathbf{y}) - 1)^2] + \frac{1}{2} E_{\mathbf{x}\sim p(\mathbf{X})}[(D_\phi(G_\theta(\mathbf{x})))^2], \tag{3.1}$$

and the generator competes against the discriminator by having the objective function

$$\min_\theta \frac{1}{2} E_{\mathbf{x}\sim p(\mathbf{X})}[(D_\phi(G_\theta(\mathbf{x})) - 1)^2]. \tag{3.2}$$

The discriminator learns a decision boundary between real and fake samples and although some fake samples might be correctly classified, Equation 3.1 penalises samples further away from the decision boundary. Since ϕ is fixed when updating the generator, the the decision boundary learned by the discriminator is also kept fixed. The dependence between the generator on the discriminator for learning (Equation 3.2) encourages the generator to produce samples closer to the decision boundary. On the next iteration, this causes the discriminator to update its decision boundary closer to the real data's manifold. The process repeats, making the decision boundary to pass through the real data manifold and the generated samples closer to the manifold of real data.

Equation 3.2, in contrast to the learning objective of the discriminator, shows the generator does not have the same level of supervision as the discriminator. While although they have competing objectives, the generator improves its generation of samples, not because of the directive by a supervisor but rather, by the information provided by the discriminator. It is through the cooperation between the generator and discriminator that the generator learns the mapping function in an unsupervised manner. This enables the ability to learn the transformation that is data driven and without any a-priori knowledge of the processes that generated the two image sets.

The GAN objectives is not limited to Equations 3.1 and 3.2. Other adversarial formulations have been developed in order to minimise other divergence measures between the observed distribution and generated distribution such as the f-divergence [113], Jensen-Shannon divergence [63] or other distance metrics to have different geometric interpretations such as, and not limited to, Earth Mover distance [114] and Integral Probability Metrics [115]. Results based on the f-divergence, Jensen-Shannon divergence and Earth Mover distance were also included in experiments but produced similar results to the LSGAN.

Cycle loss

However, the transformation $G : \mathbf{X} \rightarrow \mathbf{Y}$ is ill-posed as there are infinitely many mappings, $G(\mathbf{x})$, that could induce the estimated distribution $\hat{P}(\mathbf{X})$. This means that each $\mathbf{x}$ and output $G(\mathbf{x})$ do not necessarily have to have any meaningful relationship. For example, a possible outcome is that G_θ learns to transform all $\mathbf{x} \in \mathbf{X}$, to only one particular example of $\mathbf{Y}$. This outcome is known as mode collapse where the generator learns to map several different input values to the same output point that fools the discriminator and the model is unable to make any progress in training.

Zhu et al. [35] introduced an additional loss, *cycle* loss, that constrains the mapping to be constrained to a one-to-one correspondence (bijective mapping). If we have a mapping $G : \mathbf{X} \to \mathbf{Y}$ and another mapping $F : \mathbf{Y} \to \mathbf{X}$ then G and F should be inverses of each other. To ensure this, the generators G and F are both trained simultaneously with their own adversarial loss and own parameters, $\theta_{\mathbf{1}}$ and $\theta_{\mathbf{2}}$ respectively but also adding a loss that encourages $F_{\theta_2}(G_{\theta_1}(\mathbf{x})) \approx \mathbf{x}$ and $G_{\theta_1}(F_{\theta_2}(\mathbf{y})) \approx \mathbf{y}$. The generators G_{θ_1} and F_{θ_2} are able to reconstruct the original set of images. The distance metric used to measure the reconstruction was the L_2 norm,

$$\begin{aligned} L_{cycle}(G, F) = E_{\mathbf{x}\sim p(\mathbf{X})}[\|F_{\theta_2}(G_{\theta_1}(\mathbf{x})) - \mathbf{x}\|_2]+ \\ E_{\mathbf{y}\sim p(\mathbf{Y})}[\|G_{\theta_1}(F_{\theta_2}(\mathbf{y})) - \mathbf{y}\|_2]. \end{aligned} \tag{3.3}$$

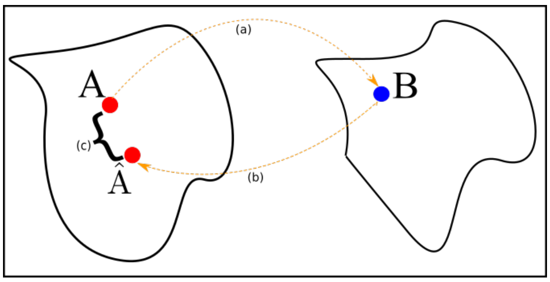


Figure 3.2: (a) Image A is mapped into the manifold of scanner set B through a a convolutional neural network (generator). **(b)** This image is then transformed back to the original manifold to reconstruct the original image using a different CNN. **(c)** The original and reconstructed image is compared using some distance metric (e.g. L_1 or L_2-norm).

Full objective

The model contains two pairs of GANs, with each generator learning the respective mapping functions $G : \mathbf{X} \to \mathbf{Y}$ and $F : \mathbf{Y} \to \mathbf{X}$. Each generator will have their respective discriminators, D_{ϕ_1} and D_{ϕ_2}, where D_{ϕ_1} will discriminate between $\mathbf{x} \in \mathbf{X}$ and samples from F_{θ_2} and conversely, D_{ϕ_2} will distinguish between $\mathbf{y} \in \mathbf{Y}$ and the output of G_{θ_1}. The objective function of G_{θ_1} and D_{ϕ_2} is given respectively as

$$\min_{\theta_1} E_{\mathbf{x}\sim p(\mathbf{X})}[(D_{\phi_2}(G_{\theta_1}(\mathbf{x})) - 1)^2] + \lambda L_{cycle}(G_{\theta_1}, F_{\theta_2}), \tag{3.4}$$

$$\min_{\phi_2} E_{\mathbf{y}\sim p(\mathbf{Y})}[(D_{\phi_2}(\mathbf{y}) - 1)^2] + E_{\mathbf{x}\sim p(\mathbf{X})}[(D_{\phi_2}(G_{\theta_1}(\mathbf{x})))^2], \tag{3.5}$$

where λ is a constant that controls the relative importance between the adversarial loss and reconstruction loss. The objective function for F_{θ_2} and D_{ϕ_1} are similarly defined.

3.3.4 Implementation

The generators and discriminators are fully convolutional neural networks. The discriminators are composed of three convolutional layers to create a receptive field of overlapping patch that aims to classify whether these image patches are either real of fake. The transformations of the input consists of a succession of spatial 2D convolutions, a leaky ReLU activation function and an instance normalisation.

During training, the input distribution of each hidden layer may change after several iterations, known as internal covariate shift, due to the complicated non-linearities of the incoming neurons. The current hidden layers will have to continually adapt to these changes in the input distribution hence could slow down convergence. Instance normalisation attempts to rectify this by normalising the inputs to each hidden layer so that their distribution during training remains fairly constant [116] which improves convergence of training. In regards to the choice of activation function, the leaky ReLU activation function was used as it was found to have the best qualitative performance except in the last layer of the discriminators where no activation function was used.

The generators follow a U-NET architecture that has been widely popular in other biomedical applications [70]. The proposed U-NET contains three convolutional down sampling layers, reducing the dimensionality of the image by a factor of eight. The downsampling layers are composed of one 3×3 convolution with stride of 2, followed by a leaky ReLU activation, instance normalisation, doubling the number of feature channels at each downsample. These layers are followed by an upsampling of the feature map followed by a 3×3 convolution ("up-convolution") that halves the number of feature channels, followed by a leaky ReLU activation and a concatenation with the correspondingly cropped feature map from the downsampling path. However, the last layer of the generator uses a 1×1 convolution is used to map each feature channel and a *tanh* function that scales the output from -1 to 1, producing a new grey matter voxel map. More specific details about the architecture used is found in Table 3.2.

(a) Architecture of Generator

Layer	Layer Type	No. of Filters	Stride	Instance Norm	Activation
1	Conv	32	2	No	LeakyReLU
2	Conv	64	2	Yes	LeakyReLU
3	Conv	128	2	Yes	LeakyReLU
4	ConvT	64	2	Yes	LeakyReLU
5	ConvT	32	2	Yes	LeakyReLU
6	Conv	1	1	No	Tanh

(b) Architecture of Discriminator

Layer	Layer Type	No. of Filters	Stride	Instance Norm	Activation
1	Conv	64	2	No	LeakyReLU
2	Conv	128	2	Yes	LeakyReLU
3	Conv	256	2	Yes	LeakyReLU
4	Conv	1	1	No	None

Table 3.2: Architecture of GAN. **Conv**: Convolution. **ConvT**: Convolution Transpose

During training, a batch size of one sagittal slice was constructed from each scanner set. The filters of the CNN were initialised as described by Glorot and Bengio [44]. The network was trained using Adam optimisation [46] with a starting learning rate of 2×10^{-4} for the generators and discriminators. The generators and discriminators were trained concurrently; every one gradient step of the generator was taken with the discriminator parameters fixed followed by a gradient step of the discriminator, keeping the generator parameters fixed. Training was stopped when the cycle loss (Equation 3.3) failed to stop decreasing. It was found empirically that the hyperparameter, λ, in Equation 3.4 was set to $\lambda = 0.2$.

3.3.5 Regression based correction methods

The performance of the GAN correction was compared against two other popular *post hoc* correction methods: linear regression and Gaussian Process (GP) regression, which have previously been used to compensate for non-disease specific effects [98, 100, 101].

A regression model was learned to estimate the GM density for every voxel based on examples of subject-specific covariate and their corresponding GM density maps. The general linear model for the voxels is given as

$$\mathbf{y} = \beta_0 + \mathbf{X}\beta + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{3.6}$$

where $\mathbf{y}$ is a $N \times v$ matrix, where the columns represent the observed GM concentrations of each voxels and the rows are the observations of each of the N control subjects. $\mathbf{X} \in \mathbb{R}^{N \times 2}$ is the design matrix representing the subjects' scanner characteristic, coded as $\{0, 1\}$ and the intercept term. $\beta \in \mathbb{R}^{2 \times v}$ represents the effect strengths associated to the scanner for each voxel and the coefficient of the intercept. The regression parameters, β, were estimated for each voxel independently with only the control subjects to avoid the confounding of disease. The model was applied to new data, $\mathbf{x}^{(*)}$, to obtain a subject specific template, and was subtracted from the observed GM map to get a corrected image.

$$\hat{\mathbf{y}}^{(*)}_{OLS} = \mathbf{y}^{(*)} - \mathbf{x}^{(*)}\hat{\beta}. \tag{3.7}$$

where $\hat{\mathbf{y}}^{(*)}_{OLS}$ is the corrected GM map of the original, $\mathbf{y}^{(*)}$ of the test example.

The GP regression correction method is analogous to Equation 3.7.

$$\hat{\mathbf{y}}^{(*)}_{GPR} = \mathbf{y}^{(*)} - (\mathbf{k}^{(*)}_{\theta})^T \mathbf{K}^{-1}_{\theta}\mathbf{y}. \tag{3.8}$$

$\hat{\mathbf{y}}^{(*)}_{GPR}$ and $\mathbf{y}^{(*)}$ are the corrected and original images respectively. $\mathbf{K}_{\theta}$ is the covariance kernel matrix of the training examples with the elements corresponding to the output of the kernel function $k_{\theta}(\mathbf{x}_i, \mathbf{x}_j)$, for $i, j \in \{1, ..., N\}$. The coefficients of the regression, $\mathbf{k}^{(*)}_{\theta}$, are the kernel function values of the test example with all the training examples. The kernel used was similar to [101] where the covariance between the input images $\mathbf{x}_i$ and $\mathbf{x}_j$ was

$$k_{\theta,\sigma}(\mathbf{x}_i, \mathbf{x}_j) = \theta_1^2 \exp(-\theta_2^2(\mathbf{x}_i - \mathbf{x}_j)^2) + \theta_3^2 + \theta_4^2(\mathbf{x}_i)^T\mathbf{x}_j + \sigma^2\delta_{ij}, \tag{3.9}$$

where $\theta_k, k = \{1, ..., 4\}$ and σ are scalar model hyperparameters, and δ_{ij} is the delta function; one if $i = j$ and zero, otherwise. The optimal hyperparameters were determined by maximising the marginal likelihood function.

3.3.6 Support vector machine classification

Each correction method in this report (GAN, GP regression, linear regression) was evaluated by the improvement of a learned supervised classifier in a range of problems such as scanner, gender and disease classification. This evaluation method was used because of the lack of ground truth; there were a limited number of subjects who were scanned across the two centers in similar conditions ($n = 11$, see Experiment 4: Reconstruction), which was insufficient to fully appraise our correction methods.

A popular technique for the classification of high dimensional neuroimaging data is the Support Vector Machine (SVM). It has been used for classification of many neurological diseases such as Alzheimer's Disease [117, 118], Huntington's Disease [101] and schizophrenia [119–123]. SVMs learn a decision boundary based on labeled examples by maximising the margin between training examples and minimising the norm of the solution vector $\hat{\mathbf{w}}$,

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b)) + \lambda ||\mathbf{w}||^2, \tag{3.10}$$

where the parameter $\lambda > 0$ determines the tradeoff between increasing the margin-size and ensuring that $\mathbf{x}_i$ lies of the correct side of the margin. Optimising Equation 3.10 can be rewritten as a constraint optimisation problem with a differentiable objective function in the following way, called the primal problem,

$$\begin{aligned} &\min \frac{1}{n} \sum_{i=1}^{n} \zeta_i + \lambda ||\mathbf{w}||^2 \\ &\text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, \text{ for all } i. \end{aligned} \tag{3.11}$$

The GM concentrations of each voxel was used as input for the classification. The primal solution, $\hat{\mathbf{w}}$, when using a linear SVM, is a linear combination of the input voxels and hence the spatial patterns of voxels that were relevant for the classification process can be visualised.

3.3.7 Postprocessing

PCA was used to transform the middle five sagittal slices of the normalised images into orthogonal eigenvector components, ordered according to their contribution of variation in explaining the set of slices. The first 50 components was used as features to train the supervised learning models as outlined in Section 3.3.8.

3.3.8 Evaluation methods

The effectiveness of each correction technique was assessed by the classification performance of a Gaussian kernel SVM. Accuracy, precision and recall of the learned SVM was evaluated using 10-fold cross validation after each correction method was applied to the dataset, as well as a baseline of no correction. For robust evaluation, the results reported were obtained in the following manner: for a test fold, the performance measure (accuracy, precision, recall and specificity) was computed for each of the correction methods and baseline. The difference of each measure was taken between baseline and the correction method. This was repeated for every test fold, collecting 10 sample sets for each method in each experiment. The average and standard deviation over the 10 sample sets was calculated for each method, and are the values reported. Significant differences in performance between each correction method and baseline were then compared by t-test with Dunnett's correction to control the type-I error rate at $\alpha = 0.05$.

3.4 Experiments

3.4.1 Supervised classification test of scanner

After preprocessing, the images were converted to bias-field corrected, normalised, grey matter density maps, however site-related differences still existed in this dataset.

To illustrate the confounding influence that site-related differences can have on the ability to classify images, we initially performed a disease classification on our preprocessed (but untransformed) full dataset. Our full dataset contained images from two different groups and two different scanners. A polynomial SVM indicated the diagnostic groups were only weakly separable, and the decision boundary tended to separate scanners rather than clinical groups. Figure 3.3 shows a representation of the decision-boundary. The figure shows the decision-boundary (background colour)

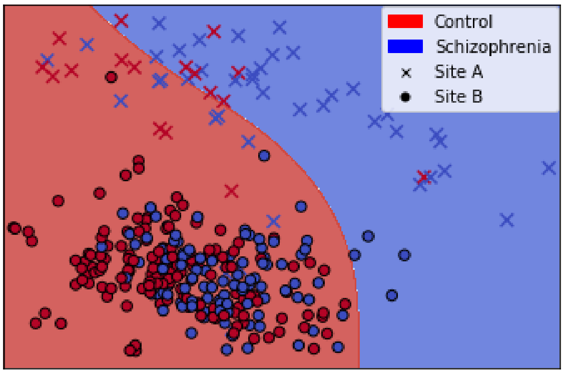


Figure 3.3: The decision boundary, plotted in 2D, learned by a polynomial SVM when classifying diagnostic groups. The background colour represents the decision boundary. The colour of points represents the true diagnostic group membership, and the shape of points represents the scanners.

tends to separate shapes representing scanner differences (crosses and circles) rather than colours representing diagnostic differences (blue vs red). In particular, the crosses and circles are well-separated to the top right and bottom left of the figure, while the blue and red circles in the bottom left are intermingled. This impairs the accuracy of the model when using to predict unseen cases and favours the prediction of the sites rather than the clinical diagnosis.

We evaluated the ability of our generative adversarial network to remove the site-related differences in our dataset. We used the mid-sagittal slice from the T1-weighted MRI of healthy subjects from site A and site B, and we merged the distribution of each image set by transforming the images from site A into images that have similar morphological characteristics as site B. Figure 3.4 shows a number of examples from the different sets and their resulting transformations. The transformed images (second row) demonstrate more consistency compared to the corresponding original images (top row). The differences between the original and transformed images, highlighted in the bottom row show significant changes in regions such as the thalamus and the brain stem.

Figure 3.5 demonstrates the changes in the mean image before and after the transformation using the GAN. The top rightmost image in Figure 3.5 shows that

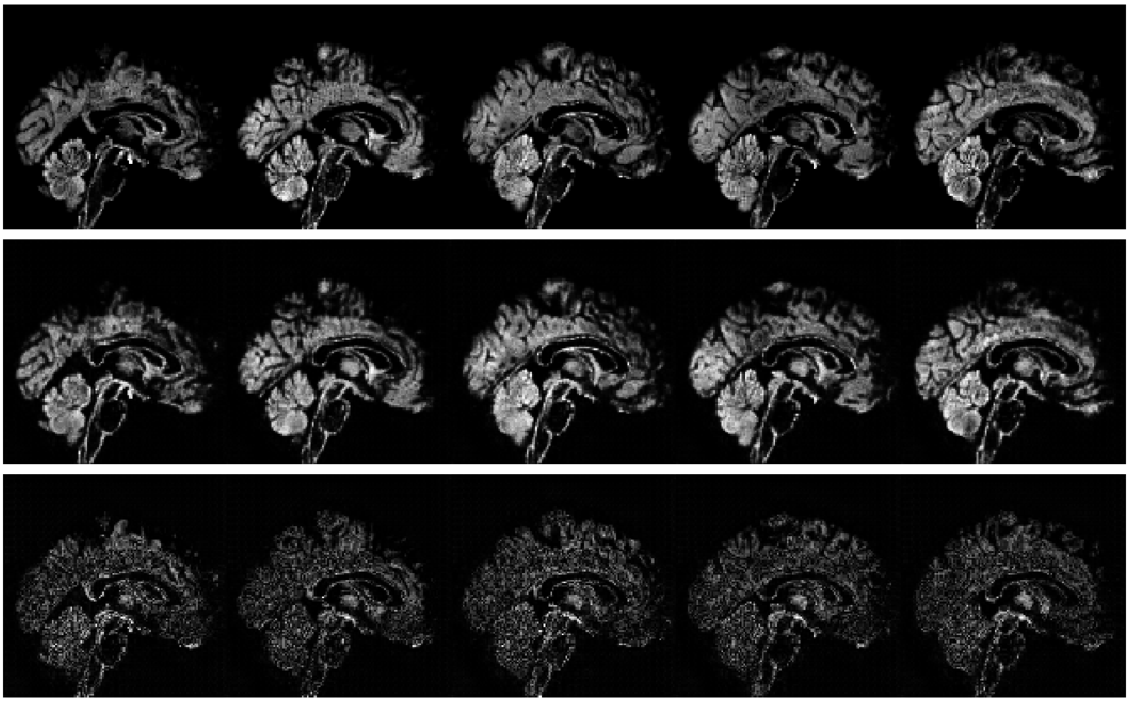
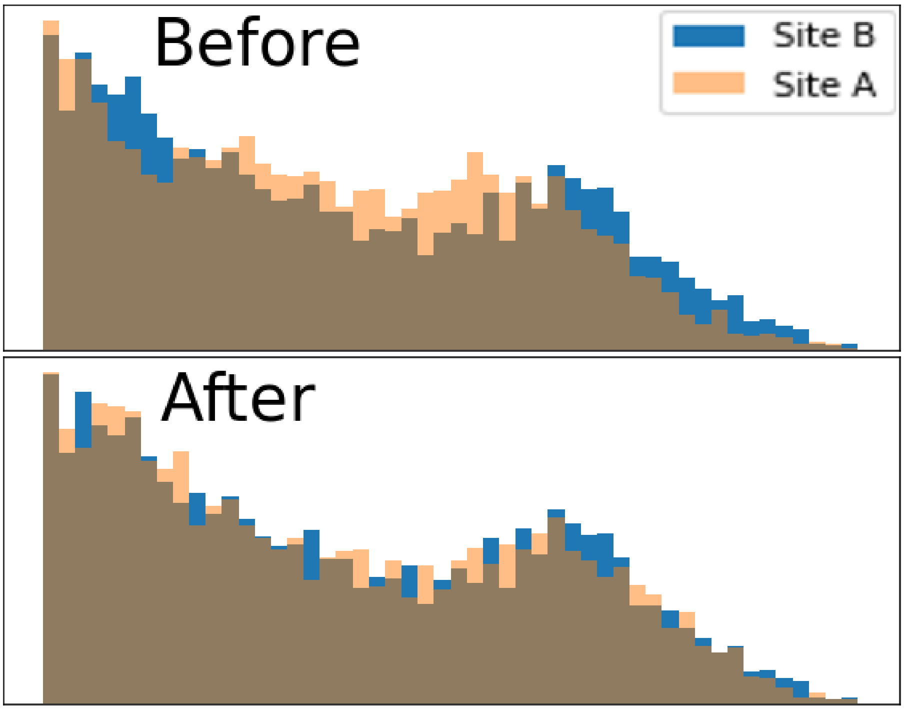


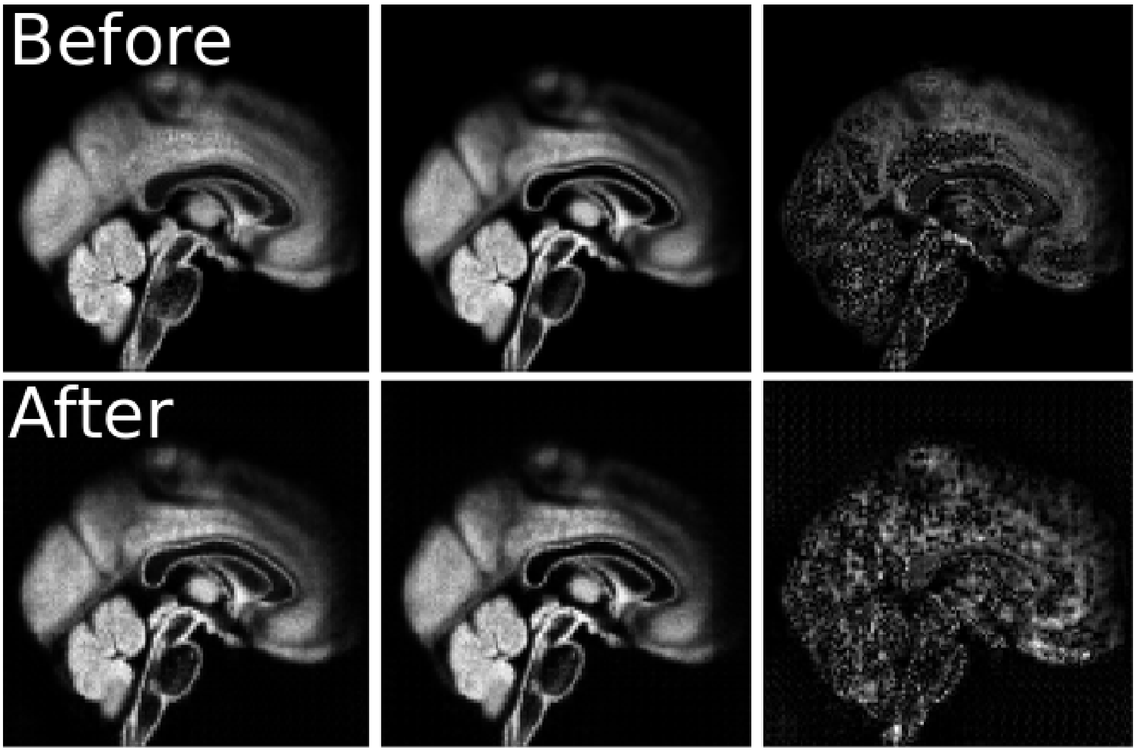
Figure 3.4: Top row: Samples of images from site A. **Second row**: The result of the transformation of images from the top row using GAN. **Bottom row**: The absolute difference between the images of first and second row.

the differences in the mean of Site A and B are particularly localized to the thalamus and the frontal lobe, however after the transformation, the differences are not concentrated to a particular area of the brain. Similarly, the GAN brings the distribution of pixel intensities between Site A and B closer to each other as shown in Figure 3.5a.

We next conducted a supervised classification test of the dataset to determine how well the images from each site were distinguishable. A Gaussian SVM model was trained using the images from healthy controls. Table 3.3 shows the performance of the classifier after different correctional techniques were applied to the healthy dataset, including linear regression, Gaussian regression, and our GAN transformation. The SVM was able to achieve close to 100 percent accuracy when discriminating between the two sites without any correction (99.3% accuracy, 99.4% precision, 99.3% recall and 100% specificity). The linear correction method produced the worst outcome as the SVM was able to distinguish between the two site images with 100%



(a)



(b)

.

Figure 3.5: Change in the mean image distributions of Site A and B, before (top rows) and after (bottom rows) transformation to a common distribution. **(a)** Distribution of pixel intensity before and after transformation. **(b)** Mean image from Site A (left) and Site B (middle) and the mean difference (right), before and after transformation.

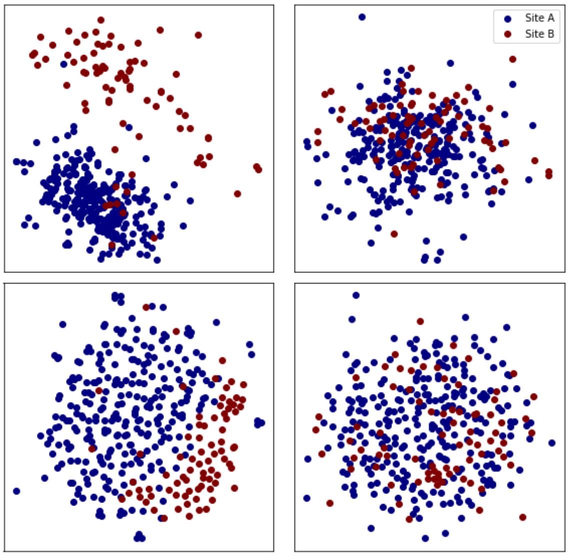


Figure 3.6: Left column: Images before transformation. **Right column**: Images after GAN transformation. **Top**: PCA visualisation of the two scanner sets. **Bottom**: a t-SNE visualisation

accuracy after application of this method. By contrast, the non-linear correction methods such as the GAN and GP regression reduced but did not eliminate, the model's ability to distinguish between the sites. This suggests that the non-linear correction methods remove or minimise the site artefacts present in our dataset, with the **gan** transformation producing the largest correction.

Table 3.3: Classification of scanners, using different correctional methods. Average difference in performance from baseline (no correction) across 10-fold cross-validation. Bold indicates the best performing in the category. Standard deviation in square brackets.

Correction method	Accuracy	Precision	Recall	Specificity
Linear regression	0.007 [0.0004]	0.006 [0.0003]	0.007 [0.0004]	0.000 [0.0000]
GP regression	-0.309 [0.0243]	**-0.476** [0.0353]	-0.309 [0.0243]	-0.049 [0.0036]
GAN	**-0.386** [0.0091]	-0.389 [0.0306]	**-0.386** [0.0091]	**-0.255**[0.0151]

3.4.2 Unsupervised classification test of scanner

We performed unsupervised learning to determine whether any unstructured information related to site differences remained in the dataset. Figure 3.6 shows a 2D visualisation of the differences between data sets before and after the transformation by the GAN, using two dimensionality reduction techniques: PCA and t-SNE [59]. t-SNE, unlike PCA, is a non-linear method that is useful for exploring local neighbourhoods and finding clusters in data (See Section 2.2.2). If data is naively pooled (left column), there is clear separation between the datasets from each site, suggesting that these site artefacts are a possible confound and will make any interpretation of results using pooled data difficult. However, after the GAN transformation (right column), such separation has vanished and the data is akin to that generated from the same distribution.

3.4.3 Classification of disease

The previous experiment demonstrated the GAN transformation method removed site-related information from our dataset on the basis of supervised and unsupervised classification methods. An important concern is whether the information loss is selective to site differences or whether other information such as that related to clinical diagnosis, is also lost. To test that, we determined whether classification of clinical diagnosis was adversely affected by any of our correction methods. A Gaussian SVM was used to classify the diagnosis of the subjects as either healthy or schizophrenia. The SVM was able to achieve over 85 percent accuracy when discriminating between clinical diagnosis without any correction (87.1% accuracy, 89.1% precision, 87.1% recall and 95.7% specificity). Table 3.4 shows comparisons compared to baseline using each correction method (Linear and GP regression, and GAN transformation).

Linear regression was the only method to produce negative changes in accuracy, implying it non-selectively removed information from our dataset. On the other hand, GP regression and GAN transformation produced significant improvements in accuracy, with GAN producing the largest improvement in accuracy (3.7%) when compared to base and 1.2% compared to GP regression. The negative changes in

Table 3.4: Classification of disease, using different correctional methods. Average difference in performance from baseline (no correction) over each cross validation fold is reported. Bold indicates the best performing in the category. Negative values indicate a worse result compared to baseline. Standard deviation in square brackets.

Correction method	Accuracy	Precision	Recall	Specificity
Linear regression	-0.003 [0.0007]	0.000 [0.0005]	-0.003 [0.0007]	**0.000** [0.0010]
GP regression	0.025 [0.0010]	0.021 [0.0010]	0.026 [0.0010]	-0.042 [0.0063]
GAN	**0.037** [0.0011]	**0.028** [0.0008]	**0.038** [0.0011]	-0.043 [0.0032]

specificity after GP and GAN correction indicate there is some improvement of classification accuracy of the schizophrenia brain images at the expense of healthy brain images.

3.4.4 Classification of gender

The GAN correction appears to selectively remove information related to site differences in our dataset, without adversely affecting information related to subtle clinical differences. However anatomical differences between psychiatric groups are likely to be small, obscure and perhaps not generally representative of the morphological changes produced by our correction methods here. Furthermore, the contribution of diagnostic groups from each site in our dataset is unbalanced (e.g., see Table 3.1), and there are reasonable concerns that unbalanced sampling from confounded groups may artificially inflate classification accuracy, even after weighting for unbalanced groups [98]. To help determine the general impact of our correction methods on anatomically distinct groups, and to eliminate concerns of inflated classification accuracy due to unbalanced groups, we tested the effect of GAN correction on balanced groups. We created a dataset which balanced the group contribution from each site by randomly selecting a set of 37 male images and 37 female images from each site. Thus, we balanced both gender and site in this dataset. Male and female images from each site were then pooled together, and correction methods were applied to each dataset. We then tested whether a Gaussian SVM could classify

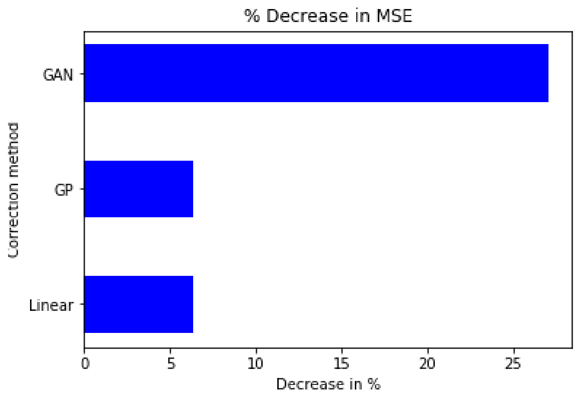


Figure 3.7: Percentage decrease in reconstruction (MSE) error against baseline for the different correction methods.

brain images by gender. On a balanced dataset, the baseline classification accuracy of the SVM (i.e., uncorrected images) was less than 70 percent (65.2% accuracy, 65.6% precision, 64.5% recall and 65.9% specificity). The results of our correction methods are shown in Table 3.5. The GAN corrected images improved accuracy by 15.8% compared to baseline whereas linear regression and GP regression produced no significant difference in the classification of gender from baseline (and on average they even reduced classification performance).

Table 3.5: Classification of gender, using different correctional methods. Reported values correspond to the average of the differences of each cross validation fold test between baseline (no correction) and the correction method. Bold indicates the best performing in the category. Negative values indicate a worse result compared to baseline. Standard deviation in square brackets.

Correction method	Accuracy	Precision	Recall	Specificity
Linear regression	-0.015 [0.0027]	-0.018 [0.0032]	-0.016 [0.0053]	-0.014 [0.0018]
GP regression	-0.033 [0.0026]	-0.036 [0.0022]	-0.025 [0.0056]	-0.041 [0.0071]
GAN	**0.158** [0.0332]	**0.130** [0.0362]	**0.211** [0.0310]	**0.105**[0.0576]

3.4.5 Reconstruction

11 subjects (5 male) had undergone MRI scans at site A and site B. This allowed us to determine how similar the reconstructed images from the different methods were to images of the same brain collected at the actual site. Images from site B were

corrected to site A and were compared to the actual images collected at site A for the selected subjects. The Mean Squared Error (MSE) between the corrected and actual image for each subject was calculated and was compared to baseline. Linear regression and GP regression performed similar to each other with a 6.35% decrease in error. The GAN correction had significant improvement over the other regression methods with a 27.02% decrease in error.

3.5 Discussion

Although combining structural MRI scans from different centres provides an opportunity to increase the statistical power of brain morphometric analyses in neurological and neuropsychiatric disorders, one important confound is the potential for site differences (scanner and MRI protocol effects) to introduce systematic errors. Thus, pooling data from different sites, scanners or acquisition protocols could make the interpretation of results difficult or even decrease predictive accuracy [119, 124]. These site specific differences are even more important with the growing popularity of open source data and automatic diagnostic systems using machine learning techniques. Although naively pooling data from multiple centers may increase sample size and intuitively, increase predictive accuracy, we found that the decision boundary learned by the classifier is heavily biased towards the separating hyperplane of the scanner differences rather than the true diagnostic label (See Figure 3.3).

We proposed a novel method using deep learning to correct (unknown) site differences and experimented with data from subjects differing in clinical diagnosis or gender. The dataset was collected at two different MRI sites with different hardware and protocols. As such, our dataset probably represents larger site-related differences than previous studies which used images acquired with similar MRI protocols [101]. Even with these large differences, we were able to remove the majority of site effects without any apparent loss in classification accuracy. These results suggest that GAN models may be a powerful method to selectively remove unwanted information from image data, without affecting the information content related to features of interest (e.g., clinical diagnosis).

The GAN transformation left intact differences related to clinical diagnosis as well as gender. Such differences are likely to vary in magnitude relative to the site-related differences the GAN removed. For instance, Voxel-Based Morphometry (VBM) and Multi-Voxel Pattern Analysis (MVPA) indicates grey matter volume differences related to schizophrenia are small, heterogenous and widely-distributed [125, 126]. By comparison, gender differences are likely larger, with fewer major points of focus, but still widely-distributed [127]. Demonstrating the selectivity of the GAN transformation against differences of varying magnitude is an important validation of the generalizability and utility of this method.

Perhaps not surprisingly, the GAN transformation produced the largest changes in the thalamus and brain stem. These regions may be more susceptible to distortions in magnetic fields and are notoriously difficult to achieve accurate image segmentation and registration during preprocessing [109]. This is partly because it has a mix of gray and white matter which cannot be easily delineated by standard preprocessing steps. An implication of the regional variations in transformation we found is that one cannot assume that preprocessing removes all site-related differences in multi-site studies, even if bias-field correction is included. However at present it is hard to do more than speculate as to why the GAN transformation produced the changes where it did.

In comparison to other learning-based approaches, one advantage of neural networks is that no features have to be hand-crafted but instead, the model learns suitable features for the transformation during training automatically [128]. In contrast to methods such as linear regression that treat voxels independently of each other, convolutional neural networks take local information into account as they are based on image patches. The fully convolutional architecture allows for a variable number of input sizes however the quality of the generation of images may change due to the fixed receptive field of the networks.

The experiments suggest that using methods such as linear regression, and in some cases GP regression (see Table 3.3-3.5) are not suitable to correct for site differences. The linear regression included an intercept term to account for mean differences between sites, yet it decreased classification accuracy when discriminating diagnostic groups and still allowed for differentiation between scanners. On the other hand, the GAN method here was able to capture the differences between scanners, making the transformations indistinguishable between the scanner sets and improve classification accuracy compared to baseline. This suggests that site-related differences are highly nonlinear that cannot be estimated using linear methods.

The small difference in performance between the GAN and GP regression when classifying diagnostic groups could be explained by the fact we only used a single sagittal slice from each brain in our dataset. A single slice would likely contain a relatively restricted amount of variance and hence represent a limit to the amount of information that can be learned from the data. The GAN correction, however, increased classification of gender significantly compared to GP regression. Figure 3.4 shows that most of the changes between original and transformed images occur around the thalamus and brain stem. Since the structural differences between gender occur in these regions [127] and the result of the transformation has improved the consistency of the GM maps in those regions across scanners, this allowed the classifier to learn a decision boundary that reflected gender differences rather than variation caused by scanner differences.

Future work is to further validate the GAN on neurological diseases such as Alzheimer's diseases that have more prominent features in MRI images as opposed to psychiatric disorders like uni-polar depression. Particular regions of the brain can be of particular interest such as the cerebral cortex and certain subcortical regions. This loss results in gross atrophy of the affected regions, including degeneration in the temporal lobe and parietal lobe, and parts of the frontal cortex and cingulate gyrus [129].

3.6 Summary

In many neuroimaging datasets contain large amounts of *unpaired* data, where examples in the dataset do not contain all modalities (i.e. they do not contain images from both scanners but rather from one). On the other hand, there is a smaller fraction of examples that contain all modalities (*paired* data) and furthermore each modality is high dimensional when compared to number of datapoints. In the next chapter, we will extend this model, which focused on using a dataset with two distinct unpaired sets of examples and learning a translation in an unsupervised manner, to a model that is able to learn in a semi-supervised fashion. This model, presented in the next section, is able to leverage a dataset contains unpaired examples but also includes paired examples to improve the stability of training and the transformation between the sets of images.

Chapter 4

Semi-supvervised Imputation of Missing MR Modalities

4.1 Introduction

Magnetic Resonance Imaging (MRI) of the brain has been used to investigate a wide range of neurological disorders and depending on the imaging sequence used, can produce different modalities such as T1-weighted images, T2-weighted images, Fluid Attenuated Inversion Recovery (FLAIR), and Diffusion Weighted Imaging (DWI). Each of these modalities produce different contrast and brightness of brain tissue that could reveal pathological abnormalities. Many of the advances in the use of data-driven models in Alzheimer's disease classification [130], brain tumour segmentation [131] and skull stripping methods [132], rely on Deep Convolutional Neural Network (DCNN)). In particular, datasets such as BraTS [34] and ISLES [23] have been focusing on the evaluation of state-of-the-art methods for the segmentation of brain tumours and stroke lesions respectively. These methods do not require the use of hand designed features and instead are able to learn a hierarchy of increasingly complex features. However, they require multiple neuroimaging modalities for high performance and improved sensitivity [133] (See Figure 4.1). Collecting multiple

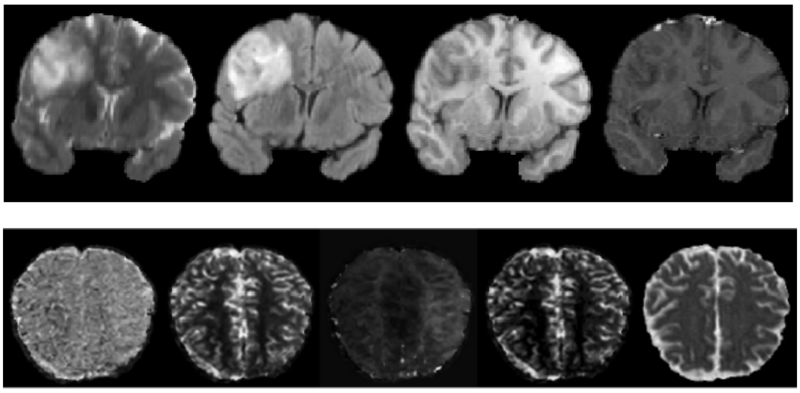


Figure 4.1: **Top**: A coronal slice of a low grade glioma (brain tumour) in the BraTS dataset in different modalities. From left to right: T2, FLAIR, T1 and T1c. **Bottom**: Axial slices of modalities of a CT perfusion scan of an ischaemic stroke lesion patient in the ISLES dataset. From left to right: mean transit time (MTT), cerebral blood flow (CBF), time-to-peak (TTP) of the residue function, CBV, ADC.

modalities for each patient can be difficult, expensive and not all of these modalities are available in clinical settings. In particular, *paired* data, where an example has all modalities present, is difficult to access, making these data dependent models more difficult to train or reduce their applicability during inference.

To ensure each modality is present, the missing modality could be imputed through a domain adaptation model where characteristics of one image set is transferred into another image set (e.g. T1-weighted to T2-weighted) that has been learned from existing *paired* examples. However, since this *paired* data is limited in the neuroimaging context, learning from examples that do not have all modalities (*unpaired* data) is valuable as this form of data is more readily available.

There has been significant interest in unsupervised image-to-image translation where *paired* training data is not available but two distinct image sets. Methods proposed by Zhu et al. [35] and Hoffman et al. [134] assume the two image collections are representations of some shared, underlying state. They use adversarial training which discriminates at the image level to guide the transformation between the domains. Furthermore, the translations between these two sets should have approximately invertible solutions and should be *cycle consistent*- where the mapping of a particular source domain to the target domain and back should yield the original source at the pixel level. Alternative methods extract domain invariant features with DCNNs and discriminate the feature distributions of source/target domains [135].

One work in recent literature that exploits the two distinct image sets of *unpaired* data, in order to improve the performance on tasks with a scarcity of *paired* data is the Cycle Wasserstein Regression GAN (CWRG) [136]. The CWRG uses the $l2$-norm as a penalty term for the reconstruction of *paired* data along with the adversarial signal and cycle-loss of the CycleGAN. However, the CWRG demonstrated its performance on ICU timeseries data and transcriptomics data and not on image data.

The proposed method, the Semi-Supervised Adversarial CycleGAN (SSA-CGAN) further extends the application of leveraging *unpaired* data and *paired* data to Magnetic Resonance Imaging (MRI) image translation, where the dimensionality of the examples is orders of magnitude larger. The method uses multiple adversarial signals for semi-supervised bi-directional image translation. Our experimental results have demonstrated that our proposed approach has superior performance compared to the CycleGAN and CWRG in terms of average reconstruction error and variance and as well as robustness to noise when evaluated using the BraTS and ISLES dataset.

4.2 Related Work

Generative Adversarial Network (GAN) have received significant attention since the work by [33] and various GAN-based models have achieved impressive results in image generation [137] and representation learning [138]. These models learn a generator to capture the distribution of real data by introducing a competing model, the discriminator, that evolves to distinguish between the real data and the fake data produced by the generator. This forces the generated image to be in distinguishable from real images.

Various conditional GAN (cGAN) have been adapted to condition the image generator on images instead of a noise vector to be used in applications such as style transfer from normal maps to images [139]. Isola et al.'s [69] work in particular, uses labelled image pairs to train a cGAN to learn a mapping between the two image domains. On the other hand, there have been significant works that have tackled image-to-image translation in the unpaired setting. The CycleGAN [35] uses a *cycle consistency loss* to ensure the forward mapping and back results in the original image. It has demonstrated success in tasks where paired training data is limited e.g. in painting style and season transfer. The Dual GAN, being inspired by dual learning in machine translation used a similar loss objective, where the reconstruction error is used to measure the disparity between the reconstructed object and the original image [140]. Unlike the previous two frameworks, the CoGAN [141] and cross-modal scene networks [142] does not use a cycle consistency loss but instead, uses weight sharing between the two GANs, corresponding to high level semantics to learn a common representation across domains.

GANs have been used in the semi-supervised learning (SSL) context as the visually realistic images generated can be used as additional training data. Salimans et al. [143] proposed techniques to improve training GANs which included learning a discriminator on additional class labels which can be used for SSL. Mayato et al. [144] modified the adversarial objective to a regularisation method based on virtual adversarial loss. The method probabilistically produces labels that are unknown to the user and computes the adversarial direction based on the virtual labels. Park et al. [145] improves upon the performance of virtual adversarial training by using adversarial dropout which maximises the divergence between the training supervision and the outputs from the network with the dropout.

GANs have been used in a range of applications in biomedical imaging such as the generation of multi modal MRI images and retinal fundus images [146], to detect anomalies in retinal OCT images [147] and image synthesis of MR and CT images [148]. Adversarial methods have also been extended to domain adaptation for medical imaging. Chen et al. [149] recently developed the Synergistic Image and Feature Adaptation framework that enhances domain-invariance through feature encoder layers that are shared by the target and source domain and uses additional discriminator to differentiate the feature distributions. Perone et al. forgoes the use of adversarial training and instead demonstrates application of self ensembling and mean teacher framework [150].

The CycleGAN has been recently applied to the biomedical field for translating between sets of data. Welander et al. [151] investigated the difference between the CycleGAN and UNIT [152] for the translation between T1- and T2-MRI modalities and found the CycleGAN was the better alternative if the aim was to generate visually realistic images as possible. McDermott et al. [136] on the other hand, tackled domain adaptation in the semi-supervised setting by proposing Wasserstein CycleGANs coupled with a l_2 regression loss function on paired data. The semi-supervised setting for this paper is similar to McDermott et al., however we propose an adversarial training signal for paired data instead of the l_2 loss. We demonstrate our method produces better reconstructions with lower variance and is more robust to noise in the context of translating between neuroimaging modalities compared to existing methods.

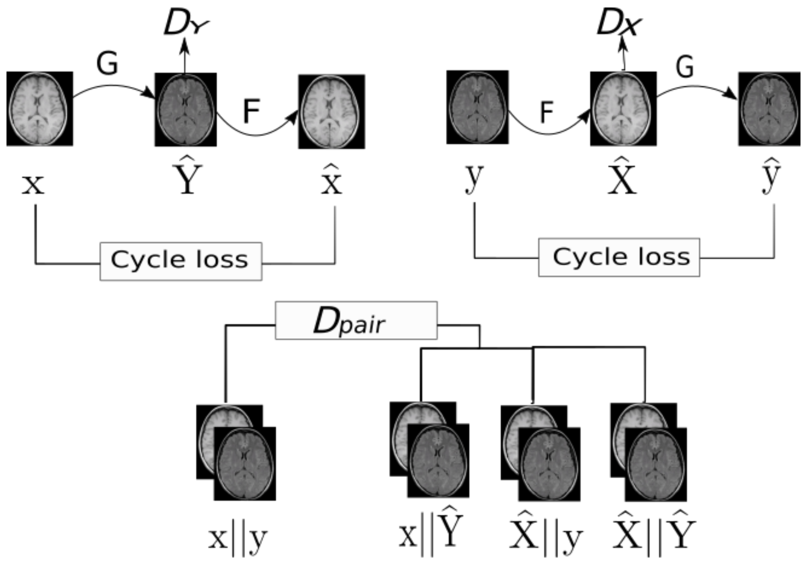


Figure 4.2: The model is composed of the CycleGAN architecture and an auxiliary discriminator which takes as input concatenated paired examples and the concatenation of generators' various transformations.

4.3 Semi-supervised Domain Adaptation with Adversarial Training

4.3.1 CycleGAN

The CycleGAN [35] learns to translate points between two domains X and Y. Given two sets of unlabeled and *unpaired* images, $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and $\{y_j\}_{j=1}^{M}$, $y_j \in Y$, two generators, F and G, are trained to learn mapping functions $G : X \to Y$ and $F : Y \to X$, where F and G are usually represented by DCNNs. Furthermore, two discriminators D_X and D_Y are trained where D_X learns to distinguish between images $\{x\}$ and $\{F(y)\}$ and D_Y discriminates between $\{y\}$ and $\{G(x)\}$. Instead of the original GAN loss, the CycleGAN trains discriminators using the least squares loss function proposed by Mao et al. [64]. For example, D_X minimises the following objective function:

$$\mathcal{L}_{D_X} = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}\left[(D_X(\mathbf{x}) - 1)^2\right] + \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y})}\left[(D_X(F(\mathbf{y})))^2\right]. \tag{4.1}$$

Conversely the generator, F, for example is trained according to the following *adversarial loss*,

$$\mathcal{L}_{F_{adv}} = \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y})}\big[(D_X(F(\mathbf{y})) - 1)^2\big], \tag{4.2}$$

as well as a *cycle-consistency loss* where reconstruction error between the inverse mapping and the original point is minimised [35],

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}\big[||F(G(\mathbf{x})) - \mathbf{x}||_1\big] + \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y})}\big[||G(F(\mathbf{y})) - \mathbf{y}||_1\big]. \tag{4.3}$$

The overall loss function for the generator is therefore given as

$$\mathcal{L}_F = \mathcal{L}_{F_{adv}} + \lambda \mathcal{L}_{cyc}, \tag{4.4}$$

where λ controls the relative strength between the adversarial signal and the cycle-consistency loss.

4.3.2 Semi-Supervised Adversarial CycleGAN

We extend the CycleGAN through the SSA-CGAN to take advantage of *paired* training data. In our scenario we have additional information in the form of T paired examples $\{\mathbf{x}_p, \mathbf{y}_p\}_{p=1}^{T}$, a subset $P \subseteq X \times Y$. We seek to take advantage of this paired information through an auxiliary adversarial network, D_{pair} (See Figure 4.2). D_{pair} takes as input, only the paired examples from P and the concatenations of the following transformations: 1) $\mathbf{x}_p$ and $\mathbf{y}_p$, 2) $\mathbf{x}_p$ and $G(\mathbf{x}_p)$, 3) $F(\mathbf{y}_p)$ and $\mathbf{y}_p$, 4) $F(\mathbf{y}_p)$ and $G(\mathbf{x}_p)$. D_{pair} attempts to discriminate between the ground-truth pairs, $\{\mathbf{x}_p, \mathbf{y}_p\} \in P$, as real and the transformation of the image and its respective real image as fake. Therefore, the paired discriminator minimises

$$\begin{aligned}\mathcal{L}_{D_{pair}} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}(\mathbf{x},\mathbf{y})}\left[(D_{pair}(\mathbf{x},\mathbf{y})-1)^2\right] + \gamma_1\mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}}\left[D_P(\mathbf{x},G(\mathbf{x}))^2\right] + \\ \gamma_2\mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}}\left[D_{pair}(F(\mathbf{y}),\mathbf{y})^2\right] + \gamma_3\mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}}\left[D_{pair}(F(\mathbf{y}),G(\mathbf{x}))^2\right],\end{aligned} \tag{4.5}$$

and F's loss is

$$\mathcal{L}_{F_{Semi}} = \mathcal{L}_{F_{adv}} + \lambda\mathcal{L}_{cyc} + \alpha\mathcal{L}_{pair}, \tag{4.6}$$

where $\mathcal{L}_{pair}$ is given as

$$\begin{aligned}\mathcal{L}_{pair} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}}\left[(D_{pair}(\mathbf{x},G(\mathbf{x}))-1)^2\right] + \mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}}\left[(D_{pair}(F(\mathbf{y}),\mathbf{y})-1)^2\right] \\ + \mathbb{E}_{\mathbf{x},\mathbf{y}\sim P_{pair}}\left[(D_{pair}(F(\mathbf{y}),G(\mathbf{x}))-1)^2\right],\end{aligned} \tag{4.7}$$

and α, λ, γ_1,γ_2 and γ_3 control the relative weight of the losses. The third loss term can be seen as further regularisation of the generators where its forward and backward transformations are pushed towards the joint distribution of X and Y.

4.4 Experiments

4.4.1 Dataset

We evaluate our method using BraTS and ISLES datasets which have been used to evaluate state-of-the-art methods for the segmentation of brain tumours and lesions respectively. BraTS utilises multi-institutional preoperative MRI scans and focuses on the segmentation of intrinsically heterogeneous (in appearance, shape, and histology) brain tumours, namely gliomas. This proposed method is trained and tested on the BraTS 2018 dataset. The training dataset contains 285 examples including 210 High Grade Glioma (HGG) cases and 75 cases with Low Grade Glioma (LGG). For each case, there are four MRI sequences, including the T1-weighted (T1), T1 with gadolinium enhancing contrast (T1c), T2-weighted (T2) and Fluid Attenuated Inversion Recovery (FLAIR). The dataset includes pre-processing methods such as skull strip, co-register to a common space and resample to isotropic $1mm \times 1mm \times 1mm$ resolution. Bias field correction is done on the MR data to correct the intensity in-homogeneity in each channel using N4ITK tool [153].

The dataset was divided as the following: 30% of examples was designated as *unpaired* examples of domain X (e.g. T2 volumes) and 30% as *unpaired* examples of domain Y (e.g. T1), 10% was designated as *paired* training examples where each example, had both T2 and T1 modalities. 10% was reserved as a held-out validation set for hyperparameter tuning and 20% was reserved to be a test set used for evaluation.

ISLES contains patients who have received the diagnosis of ischaemic stroke by MRI. Ischaemic stroke is the most common cerebrovascular disease and one of the most common causes of death and disability worldwide [154]. The stroke MRI was performed on either a 1.5T (Siemens Magnetom Avanto) or 3T MRI system (Siemens Magnetom Trio). Sequences and derived maps were cerebral blood flow (CBF), cerebral blood volume (CBV), time-to-peak (TTP), time-to-max (Tmax) and mean transit time (MTT). The dataset included images that were rigidly registered to the T1c with constant resolution of $2mm \times 2mm \times 2mm$ and automatically skull stripped [23]. The dataset includes 38 patients in total and was divided in similar proportions as the BraTS experiment regime.

Further pre-processing for each dataset included each image modality was normalised by subtracting the mean and dividing by the standard deviation of the intensities within the volume and rescaled to values between 1 and -1. The volumes were reshaped to 240×240 coronal and 128×128 axial slices for the BraTS and ISLES dataset respectively. This resulted in an average of 170 slices per patient for the BraTS dataset and 18 slices per patient in ISLES.

4.4.2 Implementation

Network Architecture: The generator network was adapted from Johnson et al. [155] and Zhu et al. [35]. The network contains two stride-2 convolutions, 6 residual blocks [156] and two fractionally strided convolutions with stride $\frac{1}{2}$. The single input discriminator networks is a PatchGAN. The paired input discriminator was a two stride-2 convolution layers. It used the concatenation of feature maps from the second last layer of D_X and D_Y as inputs as a form of weight sharing with the single image discriminators.

Training details: For all the experiments, we set $\lambda = 10$, $\alpha = 2$, $\gamma_1 = \gamma_2 = \gamma_3 = \frac{1}{3}$ in Equation 4.6 chosen by the performance on the held out validation set averaged across the pairs of MR modalities mentioned in Section 4.4.3. All networks were trained from scratch simultaneously using NVIDIA V100 GPU with an initial learning rate of 2×10^{-4}, weights were initialised using Glorot initialisation [44] and optimised using Adam [46] with a batch size of 1. The learning rate was kept constant for the first 100 epochs and was linearly decreased thereafter to a learning rate of 2×10^{-7}. Training was finished after 200 epochs. While standard data augmentation procedures randomly shift, rotate and scale images, the images were only augmented by random shifting during training as the volumes were normalised to the same orientation and shape due to co-registration.

4.4.3 Evaluation metrics

We evaluated the SSA-CGAN by learning a separate model for the following pairs of MR modalities: T2→T1, T2→T1c, T2→FLAIR, CBF→MTT, CBF→CBV, CBF→TTP, CBF→Tmax. For example, T2→T1 indicates the models were evaluated on the reconstruction of a T1 volume when transformed from a T2 volume. This was evaluated against the CycleGAN and the Cycle Wasserstein Regression GAN (CWRG) [136] which is currently the only other method in recent literature that combines *unpaired* and *paired* training data for translation between different modal-

	Method	T1	T1c	FLAIR
MSE	Cycle	0.0314 ±0.0006	0.5301± 0.4880	0.7072± 0.3956
	CWRG	0.7503 ±0.1687	0.4607± 0.3602	0.6145± 0.4279
	SSA-CGAN-p	0.0234 ±0.0032	0.0160 ±0.0100	**0.0147± 0.0018**
	SSA-CGAN	**0.0169± 0.0011**	**0.0102± 0.0024**	0.0177± 0.0071
MAE	Cycle	0.0608 ± 0.0041	0.4924± 0.4146	0.6231± 0.3264
	CWRG	0.6963± 0.3738	0.4564± 0.3868	0.5603 ±0.5564
	SSA-CGAN-p	0.0508±0.0037	0.0411±0.0118	**0.0390±0.0028**
	SSA-CGAN	**0.0436± 0.0011**	**0.0338± 0.0046**	0.0426± 0.0089

Table 4.1: MSE and MAE for various paired transformations averaged across five runs with one standard deviation for the BraTs dataset.

	Method	MTT	rCBV	TTP	Tmax
MSE	Cycle	0.1280± 0.1603	0.2437± 0.3111	0.0616± 0.0017	0.1887± 0.1565
	CWRG	0.5803± 0.2688	0.6826± 0.2604	0.5785± 0.2945	0.4825± 0.1722
	SSA-CGAN-p	0.0503± 0.0051	0.0262± 0.0017	0.0443 ±0.0085	0.0348± 0.0021
	SSA-CGAN	**0.0271± 0.0007**	**0.0202± 0.0014**	**0.0210± 0.0011**	**0.0235± 0.0041**
MAE	Cycle	0.2162± 0.1610	0.4236 ± 0.2957	0.1409± 0.0022	0.3048± 0.1939
	CWRG	0.6819±0.1240	0.7008 ±0.1478	0.5258±0.2860	0.5189±0.2800
	SSA-CGAN-p	0.1322±0.0059	0.0834±0.0029	0.1155±0.0118	0.0837±0.0048
	SSA-CGAN	**0.0947± 0.0018**	**0.0720 ±0.0043**	**0.0754 ± 0.0026**	**0.0613 ±0.0069**

Table 4.2: MSE and MAE for various paired transformations averaged across five runs with one standard deviation for the ISLES dataset.

ities. We also included in our experiments using the SSA-CGAN framework using only *paired* data, labelled SSA-CGAN-p. On the other hand, our proposed method, SSA-CGAN uses *paired* data and leverages *unpaired* data to improve learning. The hyperparameter settings for each method is similar to the training details mentioned in Section 4.4.2. For each transformation (e.g. T2→T1c) and for each method, five networks were learned, each with different initialisation of weights. These models were compared based on two quantitative metrics, the MSE and MAE averaged across the five runs and its standard deviation.

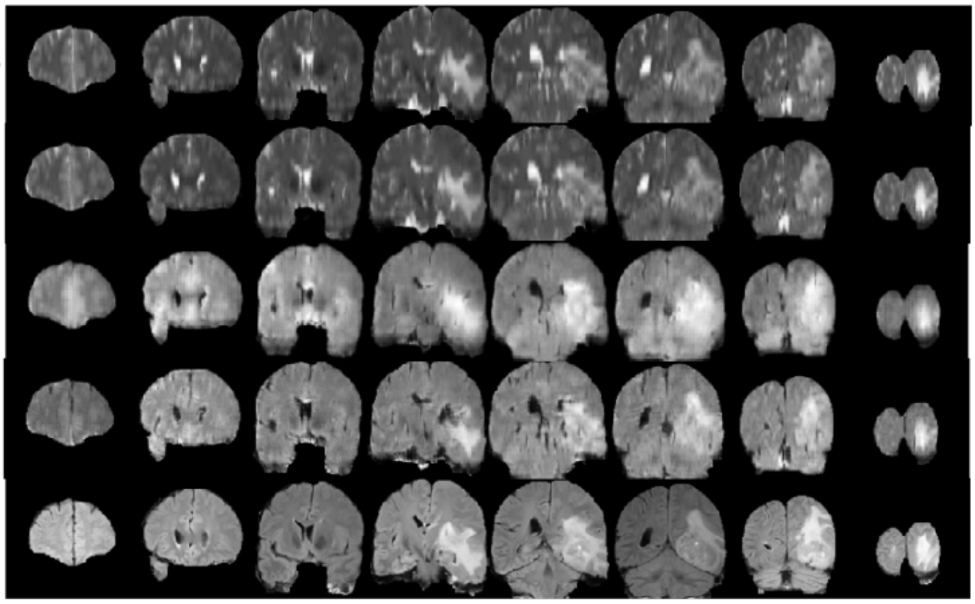


Figure 4.3: A comparison of the transformation from T2 to FLAIR.

4.4.4 Results

Results for the performance of SSA-CGAN are shown in Table 4.2. We observe that the SSA-CGAN yields from a 8.32% reduction from the CycleGAN (T2 to T1) up to a 89.6% decrease in MSE in the case of CBF to CBV with an average reduction of 33.8% and 46.0% in MAE and MSE respectively across all transformations. The consistent out-performance of our method over the CycleGAN demonstrate there is potential gains when the information from paired data points can be leveraged. This is further emphasised by the improvement over SSA-CGAN-p which has been trained using only *paired* data. By leveraging *unpaired* data during training, the SSA-CGAN produces a reduction of 18.02% and 28.16% in MAE and MSE on average when compared to SSA-CGAN-p. SSA-CGAN produces a lower MSE in most cases despite CWRG includes a loss component that minimises the l_2 norm. Furthermore, SSA-CGAN produces lower variance compared to other methods demonstrating that our method is less sensitive to different weight initialisations and improves the stability of training and convergence.

Figure 4.3 and 4.4 shows a comparison of the transformation from T2 to FLAIR and MTT to CBF respectively, of a particular chosen MR scan produced by the var-

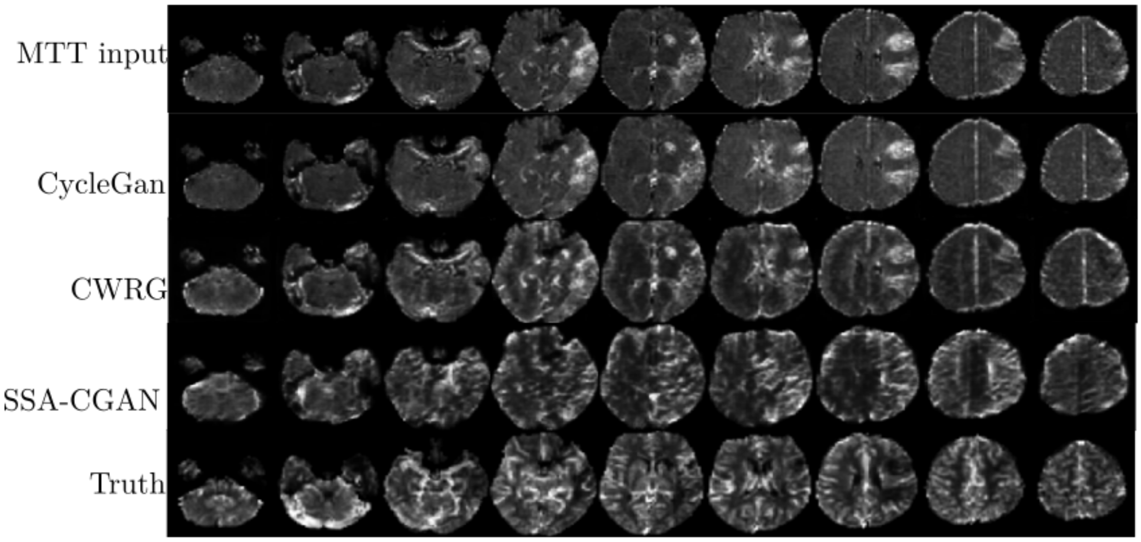


Figure 4.4: A comparison of the transformation from MTT to CBF.

ious models. The CycleGAN produces no noticeable change from the input image and the CWRG creates a smoothed version of the ground truth. This can be attributed to the MSE component of the objective function where the MSE pushes the generator to produce blurry images [157]. The additional adversarial component of our method forces the generator to synthesise a more visually realistic image. However, in Figure 4.3 the image produced does not match the pixel intensity of the ground truth and in Figure 4.4, fails to capture the high detail and edges of the CBF modality and fails to distinguish between background and low intensity areas.

4.4.5 Robustness to noise

The methods were assessed by injecting random Gaussian noise into the test data to simulate thermal noise conditions to evaluate the robustness of the models, despite not being trained on noisy examples. Various levels of noise was injected to the data, ranging from a standard deviation of 0.025 to 0.4. The predictions of the models was evaluated against the ground truth. Figure 4.6 shows the comparison between the models, with the MAE as the evaluation metric. At all noise levels, the SSA-CGAN outperforms other methods with lower variance further demonstrating the robustness of our method.

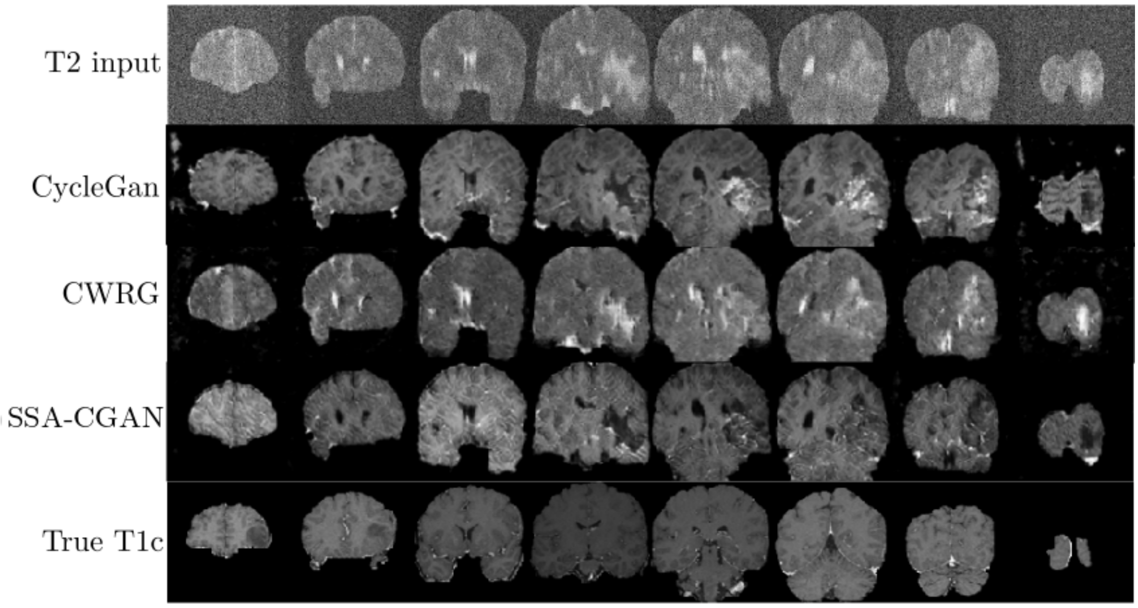


Figure 4.5: A T2 image was corrupted with Gaussian noise and was transformed to a T1c image by the various models.

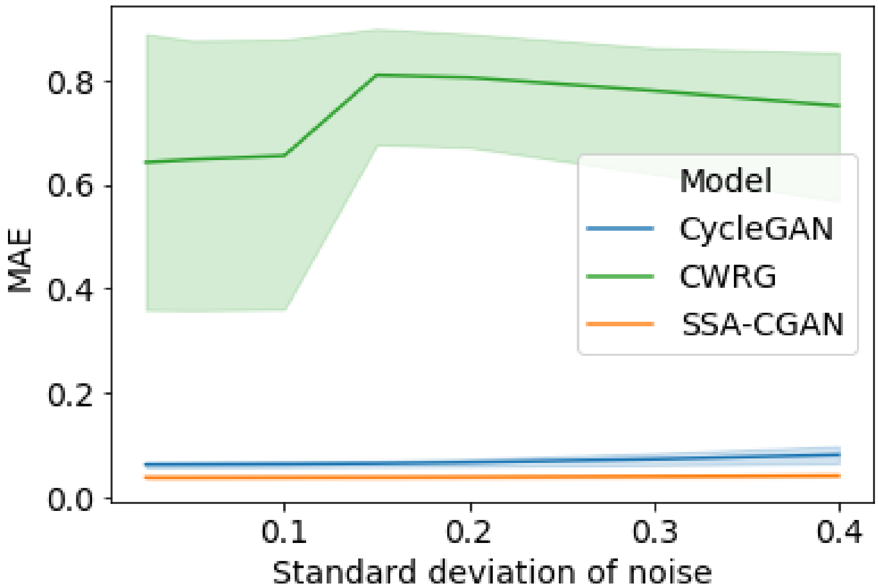


Figure 4.6: Quantitative comparison of the reconstruction error by varying the amount of random noise injected to test data.

The methods were also visually evaluated under extreme simulated thermal noise conditions by adding Gaussian noise with mean 0 standard deviation of 0.2 to the input. Figure 4.5 shows the transformation produced by a noisy input volume to the networks. The CWRG produces noise filtered version of the T2 scan and fails to perform the transformation to T1c. Our method and the CycleGAN shows robustness under the extreme scenario and fabricates successful slices. However, it fails to hide the tumour in the T2 scan (the bright spot in bottom right) in the T1c reconstruction and instead substitutes background for that tumour.

4.4.6 Limitations and Future work

This approach has several limitations. Due to the additional discriminator that distinguishes paired examples, additional computational time is required for training. Second, adversarial networks remain a very active area of research, and are known to be difficult to train and suffer issues such as mode collapse [63]. Further work would be to investigate the effect on performance when the fraction of paired examples changes and the point where the paired-input discriminator fails to be effective.

There are potential improvements to the training regime such that the networks used for the paired data can be trained initially, using the paired data which may potentially provide starting values of the parameters that can stabilise and/or improve training for the CycleGAN, i.e. the generators and discriminators trained on unpaired data.

4.5 Summary

Many state-of-the-art models in brain tissue segmentation and disease classification require multiple modalities during training and inference. However, examples where all modalities are available is limited and therefore the ability to incorporate unpaired data could be important for the adoption of these methods in clinical settings or improve existing models. Furthermore, the overall data available in limited and MRI volumes are high dimensional. The Semi-Supervised Adversarial CycleGAN (SSA-CGAN) learns translations between neuroimaging modalities using *unpaired* data and *paired* examples through a *cycle-consistency* loss, an adversarial signal for the discrimination between generated and real images of each domain and an additional adversarial signal that discriminates between the pairs of real data and pairs of generated images. The experimental results have demonstrated that SSA-CGAN has superior results in achieving lower reconstruction error and is more robust compared to all of current state-of-the-art approaches across a wide range of modality translations.

The next chapter will investigate the application of machine learning and deep learning models to another field of neuroscience: computational psychiatry. It is associated with designing decision-making tasks that discovers psychological and neural computations across groups of people, including those with underlying disease. Developing machine learning model to classify groups from these behavioural tasks can have particularly high-stakes, and thus there is an inherent need for the decision-making processes associated with machine learning algorithms to be accountable to ensure trust and transparency.

Chapter 5

Interpretable Modelling for Neuropsychological Tasks

5.1 Introduction

One main facet of computational psychiatry is the invention and administration of cognitive tasks that elicit systematically different behaviour from members of relevant populations of healthy volunteers or patient groups. For instance, bandit tasks [18], two- [96] or multi-step [158] decision-making tasks, and integrative inference tasks [159] that deliver rewards or punishments, also in the face of stress or threat, have all been employed to this effect, with relevance to bipolar disorder, OCD, depression and paranoia respectively.

An issue that arises with this approach is how to find the systematic differences between populations; a more subtle issue is whether these differences have any obvious interpretation. A sadly common dilemma in modern machine learning is that the better one can do at the former, for instance using powerful deep and/or recurrent neural networks, the harder is the latter. In some cases simple discrimination may suffice – for instance if one could generate a reliable prognosis or choose between differentially effective treatments. Particularly when these machine learning systems are integrated into society, rights and laws, such as the right of explanation in the General Data Protection Rights act introduced by the European union which gives individuals the power to demand explanations for decisions that are made by an algorithm [160] - provides a need to develops models, architectures and algorithms that incorporate a notion of interpretability.

There are currently two main approaches to interpretability in the literature. One involves the construction of process models [often drawn from reinforcement learning; 161] whose parameters, such as reward sensitivities, learning rates, prior expectations about stimuli or outcomes or the degree of reliance of choices on different decision-making systems are independently meaningful. The trouble with these models is that they can be incomplete or inaccurate. For instance, a recent study of bipolar patients playing a bandit task found that they exhibited forms of anti-reinforcement learning (switching following receiving reward) and perseveration which would not normally have been afforded appropriate parametrisations [18]. Alternatively, powerful RNNs have been used to capture behaviour more completely [18], and even, in an auto-encoder structure, to reduce the behaviour of individuals to coordinates in a self-supervised, low-dimensional, implicit parameter space [162]. However, one of the horns of the competence/interpretation dilemma above remains.

The second, more data-driven, approach to interpretability is to define transparent summary statistics. For instance, in the bandit and two-step tasks, this might be the probability of repeating an action after receiving a reward [e.g., 94, 163]. These statistics can offer very useful characterisations of tasks, but they are typically defined manually, and offer only very partial views over complex datasets.

Here, we propose a third approach that exploits and extends recent advances in interpretable forms of general machine learning by combining RNNs and prototype learning methods. This method learns for itself a prototype subsequence of behaviour for each group of subjects which characterises their overall decision-making, and simultaneously learns to classify subjects according to the similarities between these group prototypes and subsequences of the subject's own choices. The way that the comparator subsequence is chosen is also learned. These forms of learning take place concurrently. By this means, we (1) classify subjects into groups in an interpretable way by finding 'witness' subsequences in the behaviour of each, (2) extract short subsequences from each group which exemplify the whole group behaviour. The use of deep-learning for determining the choice of subsequence, is novel; as is the application to sequential behavioural data.

Through a set of experiments, we show that it is possible to gain more interpretability whilst not giving up on classification performance. We validate the framework using synthetic data and also show that when applied to the behaviour recorded from healthy subjects and patient with bipolar disorders, the model is able to extract signature behaviours of each group. The framework, therefore, offers a novel method for behavioural data analysis and may find applications in different areas of behavioural analytic, decision-neuroscience and computational psychiatry.

5.2 Related Work

Deep learning has been proven to be an effective way of modelling human choices in decision-making tasks and its relationship to psychiatric disords. For example, Dezfouli et al. [18] used RNNs to predict the next action that a subject will take in a decision-making task and thus learned to imitate the processes underlying subjects' (that were characterised as either uni-polar or bipolar depression or healthy controls) choices and their learning abilities in a two armed bandit task. These models improved upon baseline models that relied on reinforcement learning. In [164], RNNs jointly fitted to the behavioural and f-MRI data so that the model's internal state was related to neural activity and at the same time, the model's output predicted the next action of the patient. Lastly, in order to interpret differences in behaviour between groups, Dezfouli et al. [162] trained a encoder-decoder model to map the behaviour of subjects into a low-dimensional latent space. This model was trained using a loss function that included the maximum mean discrepancy and the KL divergence to ensure that the latent dimensions were informative and disentangled. As such, through their experiments found that one of the latent dimensions which represented oscillatory behaviour (subject's switched between two actions) was lower in the bipolar group than healthy controls.

Two major approaches dominate the large and growing body of work on interpretable neural networks: (1) *post hoc* methods, which derive explanations from already trained models, (2) models with inherent interpretability, in the sense that the information used by the model to make its decisions is made transparent during the process of decision-making so that it can be directly understood.

Approaches in the former category include visualising hidden states of the networks, [165, 166], extracting the importance of the features they create [167–169], and learning surrogate models that are inherently interpretable, but approximate the predictions of the target networks [32, 170]. However, *post hoc* explanations rarely capture the nuances of a model's decisions, and so can be hard to employ in critical settings, such as clinical applications [171].

To achieve inherent interpretability, one suggestion is to employ an attentional mechanism. This produces importance weights which show how strongly different parts of an input or input sequence are exploited when making a prediction. Attentional mechanisms have generally been used in tasks involving natural language [172–175]; in RETAIN [176], attention is used to highlight features of the input that are important for clinical predictions. However, just knowing the parts of the input that a classifier incorporated in its decision does not fully illuminate the reason behind the decision, and so the extent of interpretability is inherently limited.

A different form of inherent interpretability is offered by *prototype* learning, in which predictions are formed by comparing new inputs with few exemplar cases. Li et al. [177] proposed a network architecture that includes a *prototype layer*, where each unit of the layer resembles an encoded training input which is then used for case-based reasoning for classification. In order to visualize these encoded prototypes in the original input space, they train the network with a decoder. By contrast, *ProtoPNet* [178] does not require a decoder, but rather is able to dissect a given input image into its prototypical parts and combine this evidence for classification.

Ming et al. [179] extended these ideas to apply them to modelling sequential data using RNNs. In order to summarise the prototype sequences so that they could be comprehended by users, Ming et al. used beam search (a greedy breadth-first search algorithm) after learning the prototypes to find the set of subsequences that is closest to the prototypes. However, the classification of new sequences did not depend on these subsequences and these discovered subsequences may not be relevant to classification. Thus, this method is more akin to a *post-hoc* method of interpretability with respect to the subsequences.

Our method extends these prototype learning ideas but summarises an entire sequence by extracting a subsequence that is important to its classification, whilst at the same time producing representative subsequences for each of the populations that are being discriminated.

5.3 Interpreting Neuropsychological Tasks with Prototypical Networks

The proposed model can be decomposed into three main components: an encoding and attentional mechanism that, given the input sequence for each subject, outputs a relevant subsequence of that input; an encoder neural network that maps the extracted subsequence to a lower-dimensional space; and a classification mechanism that takes the encoded subsequence and classifies it based on its similarity to a set of learned prototypes in the embedding space. Figure 5.1 shows an overview of

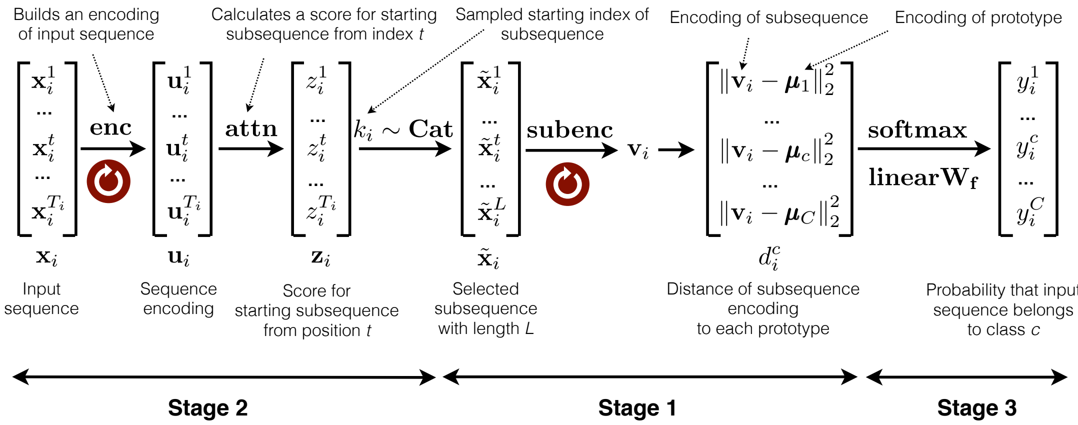


Figure 5.1: An overview of the architecture of the model and the different training stages of the network. The input sequence $(\mathbf{x}_i^t)_{t=1}^{T_i}$ goes through a recurrent sequence encoder, **enc**, followed by an attention layer, **atten**, which picks a starting index k_i. This index is used to extract the subsequence $\tilde{\mathbf{x}}_i = (\mathbf{x}_i^t)_{t=k_i}^{k_i+L-1}$. This subsequence goes through another recurrent encoder, **subenc**, followed by a prototype layer, **prot**, that measures the (dis)similarity between the embedded subsequence and the *prototypes* of each class. Eventually, these dissimilarity values go through a linear layer followed by a softmax function to compute the estimated class probabilities. The stages listed underneath represent the order in which the various parts of the network are trained. **Cat** stands for **Categorical**. Symbol ⟳ represents recurrent neural network layers.

the architecture and Section 5.3.1 provides more details about each of its various components. Training is described in Section 5.3.2.

Definitions: Let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ be a dataset of N labelled data points. For each data point i, $\mathbf{y}_i \in [0,1]^C$ represents the one-hot encoding of the class label (for instance, the patient group to which subject i belongs), and $\mathbf{x}_i = (\mathbf{x}_i^t)_{t=1}^{T_i}$ represents a sequence of length T_i. Each element of this sequence is a d-dimensional real vector, $\mathbf{x}_i^t \in \mathbb{R}^d$, which for example includes information such as the action of subject i, reward, and stimulus at time t.

5.3.1 Architecture

Subsequence Extraction

This part of the model – which is mainly based on [172] – receives an input sequence $\mathbf{x}_i$, and outputs a small subsequence of it, denoted by $\tilde{\mathbf{x}}_i = (\mathbf{x}_i^t)_{t=k_i}^{k_i+L-1}$. Here, $k_i \in$

$\{1, \ldots, T_i - L\}$ is the starting index of the subsequence and is selected based on the specific input $\mathbf{x}_i$. The length of the subsequence, L, is a fixed hyper-parameter. The rest of this subsection explains how the network selects k_i.

The entire input sequence $\mathbf{x}_i$ is processed by a recurrent sequence encoder, denoted by $\mathbf{enc}(\theta_e, .)$ with parameters θ_e. On each timestep t, the encoded vector is $\mathbf{u}_i^t = \mathbf{enc}(\theta_e, \mathbf{x}_i^t)$, where $\mathbf{u}_i^t \in \mathbb{R}^m$ and $m \ll d$ (i.e., the data point is mapped to a lower dimensional space). These encoded vectors are used as inputs to an attention layer [172–174], denoted by $\mathbf{atten}$, which outputs an alignment vector $\mathbf{s}_i = [s_i^1, \ldots, s_i^{T_i}]^\top \in \mathbb{R}^{T_i}$. Each $\mathbf{s}_i^t \in \mathbb{R}$ can be thought of as a score which represents how representative the subsequence $(\mathbf{x}_i^t)_{t=k_i}^{k_i+L-1}$ would be of the whole sequence $\mathbf{x}_i$. We describe below how $\mathbf{s}_i$ is computed; but, for the moment, consider it to be given. These scores are mapped into a probability vector $\mathbf{z_i} = [z_i^1, \ldots, z_i^{T_i}]$ using a softmax:

$$z_i^t = \frac{\exp(s_i^t)}{\sum_{t'=1}^{T_i} \exp(s_i^{t'})}, t = 1, 2, ..., T_i, \tag{5.1}$$

and an index $k_i \sim \mathbf{Categorical}(\mathbf{z}_i)$ is sampled from the corresponding categorical distribution. This determines the starting index of the subsequence $\tilde{\mathbf{x}}_i = (\mathbf{x}_i^t)_{t=k_i}^{k_i+L-1}$ which is the input to the prototype layer.

The scores $\mathbf{s}_i = [s_i^1, \ldots, s_i^{T_i}]^\top$ are computed via a one layer "additive attention" mechanism [172]:

$$s_i^t = \mathbf{V}_a^\top \tanh(\mathbf{W}_a[\mathbf{u}_i^t; \mathbf{u}_i^{T_i}]), \tag{5.2}$$

depending on the encoded vector at time t, $\mathbf{u}_i^t$, and the final encoding vector $\mathbf{u}_i^{T_i}$, where $\mathbf{W}_a$ and $\mathbf{V}_a$ are parameters. We denote by $\mathbf{\Theta}_e = \{\theta_e, \mathbf{V}_a, \mathbf{W}_a\}$ the set of all parameters of the subsequence extractor.

Prototype Matching

The next phase of processing encodes the representative subsequence $\tilde{\mathbf{x}}_i = (\tilde{\mathbf{x}}_i^t)_{t=1}^{L}$ into a vector $\mathbf{v}_i$, and then calculates a vector $\mathbf{d}_i \in \mathbb{R}^C$ that quantifies the dissimilarity of the input to the C prototypes of the classes.

More specifically, $\tilde{\mathbf{x}}_i$ is input to a recurrent (sub)sequence encoder,**subenc**$(\theta_{\mathbf{s}}, .)$. Similar to **enc**, **subenc** is also a GRU-based encoder [180]. However, while **enc** is specialised at encoding longer sequences for the purpose of subsequence extraction, **subenc** is tasked with encoding shorter (sub)sequences in a way that allows for immediate prototype matching. The output of **subenc** is $\mathbf{v}_i$, which corresponds to the last output of the recurrent encoder.

We then follow [178, 179]. The prototype layer, **prot**, is parameterised by vectors $\mathbf{M} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_C\}$, where C is the number of classes[1] and each prototype $\boldsymbol{\mu}_c \in \mathbb{R}^b$ has the same dimensionality as $\mathbf{v}_i$. **prot** calculates the dissimilarity of the input $\mathbf{v}_i$ to each prototype $\boldsymbol{\mu}_c$ defined by the squared Euclidean distance, $d_i^c = \|\mathbf{v}_i - \boldsymbol{\mu}_c\|_2^2$.

Sparse Linear Classification

The final part of the architecture turns the dissimilarity vector $\mathbf{d}_i$ into class probabilities $\hat{\mathbf{y}}_i = \{\hat{y}_i^1, \ldots, \hat{y}_i^C\}$. To achieve this, the dissimilarity vector is passed through a linear layer (**linear**; $\mathbf{o}_i = \mathbf{W}_f \mathbf{d}_i$) and then a softmax layer to produce class probabilities, $\hat{y}_i^c = \exp(o_i^c)/\sum_{j=1}^{C} \exp(o_i^j)$. $\mathbf{W}_f$ is a $C \times C$ weight matrix parameter which is trained with a sparsity constraint (see the next section), which is required to ensure that all the class prototypes are meaningful and interpretable (see Section 5.4.2).

5.3.2 Training algorithm

Following [170], the model was trained iteratively until convergence. Each iteration consists of three sequential stages: (1) training the subsequence encoder, **subenc**, and the prototype layer, **prot**, to classify sequences, thereby generating prototypes that are helpful in classification (in the first iteration, since the subsequence extractor is not yet trained, we use randomly sampled subsequences as the input to **subenc**), (2) training the sequence encoder, **enc**, and the attention layer, **atten**, to extract

[1]While the number of prototypes could be greater than C in general, choosing C prototypes helps in providing more interpretable prototypes that correspond to each group/class

meaningful subsequences, and (3) training the final fully-connected layer, **linear**, to improve the overall classification performance using the complete input sequences. In each training stage, we keep the *other* parameters of the network fixed. Given that the subsequence index k_i is sampled, conventional gradient-based optimisation methods cannot be used to train the network end-to-end. Instead, we propose a novel application of a policy gradient method.

Stage 1

The aim of the first training stage is to learn the parameters of **subenc**—namely θ_s—and the prototype vectors, $\mathbf{M} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_C\}$ keeping the other parameters fixed, including $\mathbf{W}_f$. We minimise the sum of the cross-entropy loss, which penalises misclassification of the training data, and a regularisation term that ensures prototypes remain distinct [179]:

$$\min_{\theta_s, \mathbf{M}} \mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i) + \alpha_1 \mathcal{R}_\lambda(\mathbf{M}), \tag{5.3}$$

where $\mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \frac{1}{n}\sum_{i=1}^{N} \mathrm{CrsEnt}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ and $\mathcal{R}_\lambda(\mathbf{M}) = \sum_{i=1}^{C}\sum_{j=i+1}^{C} \max(0, \lambda - ||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2)^2$. λ is the threshold that determines whether or not the prototypes are close in embedding space. This ensures that the prototypes are distinct from each other and prevents the model from collapsing into learn a single prototype.

Critically, during training, at every M epochs, since the prototypes $\boldsymbol{\mu}_c$, may not correspond to a subsequence in the training data, we project them onto the closest subsequence encoding, as described by the rule [178, 179]. This step updates $\boldsymbol{\mu}_c$ by

$$\boldsymbol{\mu}_c^{\text{new}} \leftarrow \underset{\mathbf{v}_i \in \mathbf{subenc}(\tilde{\mathcal{X}})}{\arg\min} \left\| \mathbf{v}_i - \boldsymbol{\mu}_c^{\text{old}} \right\|_2, \tag{5.4}$$

where $\tilde{\mathcal{X}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^{N}$ is the set of all subsequences that are fed to **subenc**. This ensures that each prototype corresponds to an observed subsequence and is meaningful and interpretable.

Note that in the first iteration (i.e., the first time stage 1 is run), the subsequence extractor is not trained yet; therefore, we select the subsequences uniformly at random from the input sequences. In later iterations, subsequences are chosen based on the output of the attention layer from the previous iteration. We also initialise the weights of the last fully-connected layer, $\mathbf{W}_f$, in the same manner as [178]: the weights that connect each prototype $\boldsymbol{\mu}_c$ to the logit corresponding to class c are set to 1.0, and all other weights are set to -0.5. This allows for subsequences that are similar to a particular prototype, $\boldsymbol{\mu}_c$ to increase the probability of being classified to class c and conversely, decrease the probability of being classified to the other classes.

Stage 2

After learning the embedding of subsequences and the prototypes that are appropriate for classification, the next stage of training involves training a model that will extract subsequences from the original sequence that approximates the prototypes as closely as possible. It is necessary to train this stage separately because of the stochastic sampling associated with the choice of index k_i. This is treated as a Reinforcement Learning (RL) problem where the model (agent) must find an optimal behavioural strategy that finds the index k_i for a sequence that maximises the negative cross-entropy loss (Equation 5.3) (the reward).

To solve this problem, we use a policy gradient method to set the parameters $\boldsymbol{\Theta}_e$ of the sequence encoding policy $\pi_{\boldsymbol{\Theta}_e}(a|\mathbf{x})$ realised by **enc** and the attention layer, **atten**. We use the REINFORCE algorithm [181] with an entropy term. Therefore, the objective function that is maximised is given as:

$$J(\boldsymbol{\Theta}_e) = \mathbb{E}_\pi[-(\mathrm{CrsEnt}(\mathbf{y}_i, \hat{\mathbf{y}}_i) - \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})) \ln \pi_{\boldsymbol{\Theta}_e}(a_i|\mathbf{x}_i)] + \alpha_2 \mathcal{H}(\pi_{\boldsymbol{\Theta}_e}(\cdot|\mathbf{x}_i)), \quad (5.5)$$

where $\mathcal{H}(\cdot)$ is the entropy measure and α_2 controls how important the entropy term is, known as the temperature parameter [182]. The entropy maximisation leads to policies that (1) explore more; and (2) assign options with equal probability to be chosen if there are multiple options that seem to be equally good. The cross entropy loss averaged across the training set, $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ is subtracted from the reward as a baseline term to reduce the variance of the gradient updates. Once a subsequence is chosen according to $\pi_{\mathbf{\Theta}_e}(a|\mathbf{x})$, the evaluation of the cross entropy is conditionally independent of $\mathbf{\Theta}_e$ as the the parameters of the other sections of the network are fixed and therefore $J(\mathbf{\Theta}_e)$ will have valid gradients enabling the use of gradient based methods to optimise Equation 5.5.

Stage 3

The weights of the final full-connected layer, $\mathbf{W}_f$, are trained to improve the accuracy of the model without changing the sampling model, embedding vectors and prototypes. While only one meaningful prototype is required for a two-class classification (one prototype for class c, one prototype that is *not* class c), we instead require all the prototypes to be meaningful. Following [178], we also include a L_1 regularisation term on the weights that encourages the prototypes to represent their respective classes rather than the difference between the classes. The final optimisation objective is:

$$\min_{\mathbf{W}_f} \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) + \alpha_3 \left\| \mathbf{W}_f \right\|_1 . \tag{5.6}$$

The sparsity encourages the model away from the *negative* reasoning process to the *positive* e.g. "This is a subsequence of class c because it *is* of class c" rather than

"This is a subsequence of class c because it *is not* of class b".

Algorithm 2: Prototype Network algorithm

Input: Number of epochs, N, Number of iterations per training stage, K,
Prototype Projection update , M

Output: Model with trained parameters

Initialise **enc**,**subenc**, **prot**,**atten** layers using Xavier initialisation.

Initialise **linear** layer described in **Stage 1**.

for *i=1,2,...,N* **do**
    Stage 1: **for** *k=1,2,...,K* **do**
        Update **subenc** parameters, θ_s and prototype vectors $\mathbf{M}$ using Equation 5.3
        if *k%M == 0* **then**
            Update prototype vectors, $\mathbf{M}$ using Equation 5.4
        end
    end
    Stage 2: **for** *k=1,2,...,K* **do**
        Update **enc** and **atten** using Equation 5.5
    end
    Stage 3: **for** *k=1,2,...,K* **do**
        Update $\mathbf{W}_f$ using Equation 5.6
    end
end

5.4 Experiments

In this section, we evaluate the model on two synthetically generated and two real-world datasets. We assess the method in two respects: (i) classification performance compared with baseline models, which are GRU and Attention RNN [172, 173]; (ii) interpretability of the extracted prototypes and subsequences. We also performed ablation studies to investigate the role of the different components of the architecture and the length of the subsequence, L (Section 5.4.3).

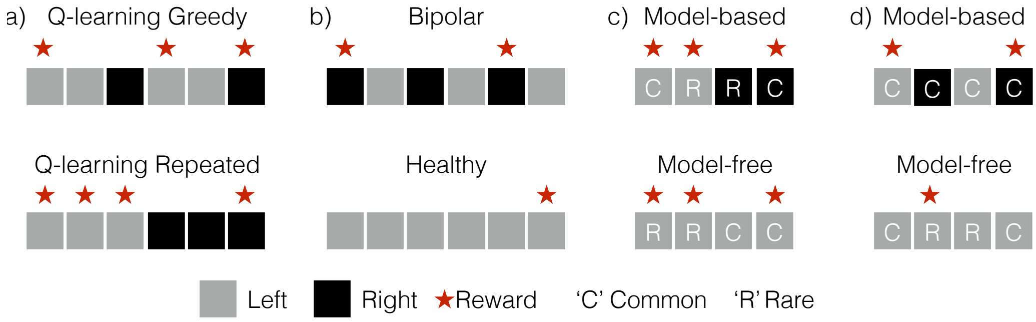


Figure 5.2: Prototypical subsequences from each of the experiments produced by our method. The squares show the actions chosen by the agent at each timestep. The red star shows whether a reward was received on the trial. **(a)**: prototype subsequences for QLG (upper) and QLR (lower) on the bandit task. **(b)**: prototype subsequences for subjects suffering bipolar disorder (upper) and healthy controls (lower). **(c)**: prototypes for the first-stage actions of the model-based (upper) and model-free (lower) agents in the Two-Stage task. Actions in the first-stage that led to a second-stage state only 30% of the time are labelled **R**, or **C** otherwise. **(d)**: prototypes for the Two-Stage task in the absence of the linear layer (using the same graphical convention).

Evaluation Regime

Each dataset is subdivided into a training set of 70% of the cases, a validation set of 10%, used to choose hyper-parameters, and a test set of 20%. For each collection of hyper-parameter values, learning was repeated using five different seeds that determined the randomised initialisation of weights and training. The results presented are for the hyperparameter values associated with the highest validation score averaged across the five seeds, but evaluated on the test set (with attendant standard deviations). The final hyperparameter values are listed in Table 5.3.

Hyperparameter settings

The dimensionality of the prototype embedding was $b = 5$ in all the experiments. Hyperparameters in Equation 5.3 were also fixed to the following values: $\lambda = 3.0$ and $\alpha_1 = 0.01$, however d should be scaled appropriately with b. Table 5.3 shows the other configurations of our model.

Dataset	L	Dim	Dropout	α_2	α_3
Bandit	2	20	0.0	0.02	0.02
	4	20	0.0	0.05	0.05
	6	**20**	**0.0**	**0.05**	**0.02**
	8	20	0.0	0.02	0.05
BD	2	30	0.1	0.05	0.02
	4	40	0.1	0.02	0.05
	6	**30**	**0.0**	**0.02**	**0.02**
	8	40	0.1	0.02	0.05
Two Stage	2	40	0.0	0.02	0.02
	4	**20**	**0.1**	**0.08**	**0.02**
	6	30	0.1	0.08	0.05
	8	20	0.1	0.05	0.02

Table 5.3: Parameters for model architecture for various datasets in the following order: dimensionality of GRU hidden state vector, subsequence length, dropout, entropy regularisation and l_1 regularisation. The bolded settings were used to present results and graphics.

Training details

We use RMSProp [45] for all the training. We iterated training on **Stage 1**, **Stage 2** and **Stage 3** three times, where each stage was trained for 200 epochs using full batches. The learning rate was initialised to 0.01 for the first 50 epochs followed by an exponential decay at a rate of 0.015 until a learning rate of 2e-4. Training was performed on NVIDIA V100 Tensor Core GPU using TensorFlow v1.15[2].

5.4.1 Classification Performance and Interpretability

Synthetic Bandit

To illustrate that our method can learn the underlying ground truth classifications and the group-level prototypical subsequences,

We constructed a 2-armed bandit problem where the agent faces two choices, a_A (LEFT) and a_B (RIGHT). At each trial, t, the actions, LEFT and RIGHT each have a fixed probability of earning reward, $P(r_t|a_A) = 0.1$ and $P(r_t|a_B) = 0.3$ respectively.

[2]https://www.tensorflow.org/

The agent must simultaneously attempt to acquire new knowledge (called "exploration") and optimise their decisions based on existing knowledge (called "exploitation"). The agent attempts to balance these competing tasks in order to maximise their total value over the period of time considered.

The agents were simulated using SARSA(λ) similar to Equation2.35 except with only one state. Similarly, state-action values were connected to choices according to Equation 2.36.

We simulated the Q-learning agents by setting the learning rate, $\alpha = 0.3$ and the inverse temperature parameter, $\beta = 1.0$ for both classes.

Furthermore, we created two classes of behaviours for these Q-learning agents [79] and simulated their behaviour on a bandit task involving two stochastically-rewarded actions, LEFT and RIGHT. One class of agents, the Q-learning greedy (QLG) agents, chooses its action according to its Q-values, updated through SARSA but for a randomly chosen sequence of 10 trials, it chooses actions in a deterministic greedy fashion (i.e., stays with the same action after reward and switches otherwise). A second class of Q-learning *repeated* (QLR) agents employs the same parameter values as QLG, but with probability 0.1, they disregard their current Q-values and instead, choose a particular sequence of actions {LEFT, LEFT, LEFT, RIGHT, RIGHT, RIGHT}. This specific pattern of actions then will be signature behaviour of QLR. Each agent generated a sequence of 100 actions and associated rewards ($T = 100$); we generated 500 agents of each class ($N = 1000$).

We trained our method to discriminate sequences of actions and rewards from QLG and QLR agents. Figure 5.2 shows the prototypical subsequences that arose. The prototype for the QLR agent is the sequence of repeated actions where it begins with the sequence LEFT three times, and despite earning a reward in the middle of the subsequence, switches to RIGHT three times. The prototype for QLG demonstrates the reward driven behaviour as defined earlier (stay after reward and switch after no reward), and also does not include any three actions in a row, helping maximise the distance to the prototype of QLR.

Table 5.4 shows the negative log-likelihood and accuracy performance on the test sets. Our method outperforms the RNN-based classifier in terms of accuracy even though discrimination is only based on a short extracted subsequence and not the entire sequence (as for the RNN).

BD

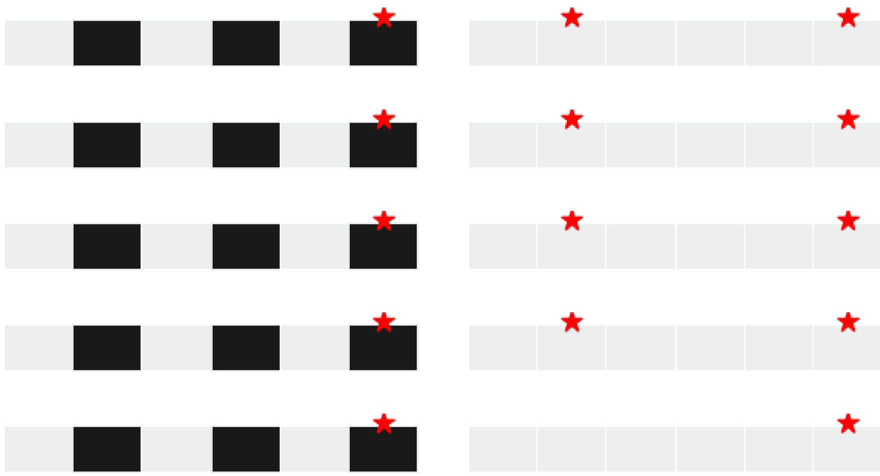


Figure 5.3: The extracted subsequences of the top 5 most confident classifications for each class in the BD test set. **Left column**: Bipolar disorder. **Right column**: Healthy.

This dataset [18] comprises behavioural data from 34 patients with depression, 33 with bipolar disorder and 34 matched healthy controls. Similar to the synthetic dataset above, subjects performed a bandit task with two stochastically-rewarded actions (LEFT and RIGHT). We focus on discriminating patients with bipolar disorder from healthy controls. Each subject completed the task 12 times using different reward probabilities for each action and each task comprised of 200 trials. The dataset thus contains $N = 12$ (sequences) $\times 67$ (participants) $= 804$.

The prototypes in Figure 5.2b show that subjects in the bipolar group have a greater tendency to switch between the actions (LEFT and RIGHT) than healthy controls, who have greater tendency to perform the same action multiple times (i.e.,

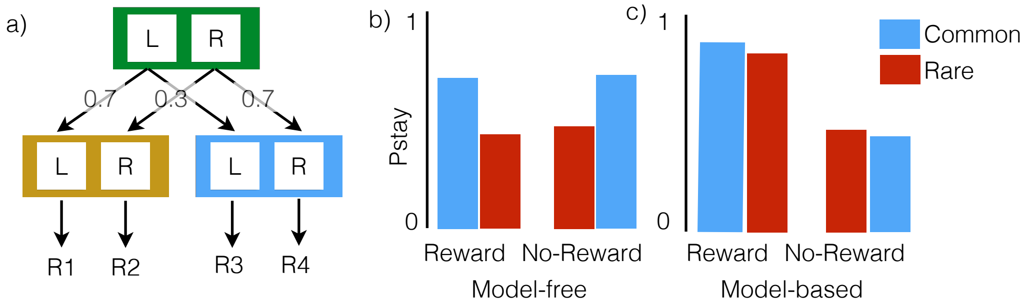


Figure 5.4: (a) The design of the Two-Stage Task along with each action's transition probabilities to the second-stage states. 'L' and 'R' refer to available actions, and R1...R4 refer to the rewards at stage 2. **(b)** The probabilities of the model-free learner to choose the same first stage action, depending on whether a reward and rare transition was observed (based on the simulated data). **(c)** The model-based learner.

to perseverate). This tendency in the bipolar subjects has already been reported in this dataset, based on intuition [18], but here was extracted without a priori knowledge about the fact that switching between the actions can differentiate between the groups. Extracted subsequences for five individuals is shown in Figure 5.3 which highlight the similarity of the subsequence to their respective class prototype.

Synthetic Two-Stage Task

As a final demonstration of the effectiveness of the method, we consider the behaviour produced by synthetic model-free (MF) and model-based (MB) reinforcement learning agents on a widely used two-stage Markov decision task [94]. This task was designed to highlight a signature difference in the behaviour of these agents according to whether they repeat the same first-stage action in the next trial, based on whether the previous led to reward and whether it involved a common or rare transition (see Figure 5.4). Again, we simulated 100 trials for of 500 agents of each type (see Supplementary for more details).

In our experiments the learning rate, $\alpha = 0.9$ and the inverse temperature parameter, $\beta = 2.0$ for both classes.

At each trial, the reward probabilities changed over time, within a limit of 0.25 and 0.75, following a Gaussian random walk with $\sigma = 0.04$. Figure 5.5 shows the reward probabilities for each of the second-stage actions.

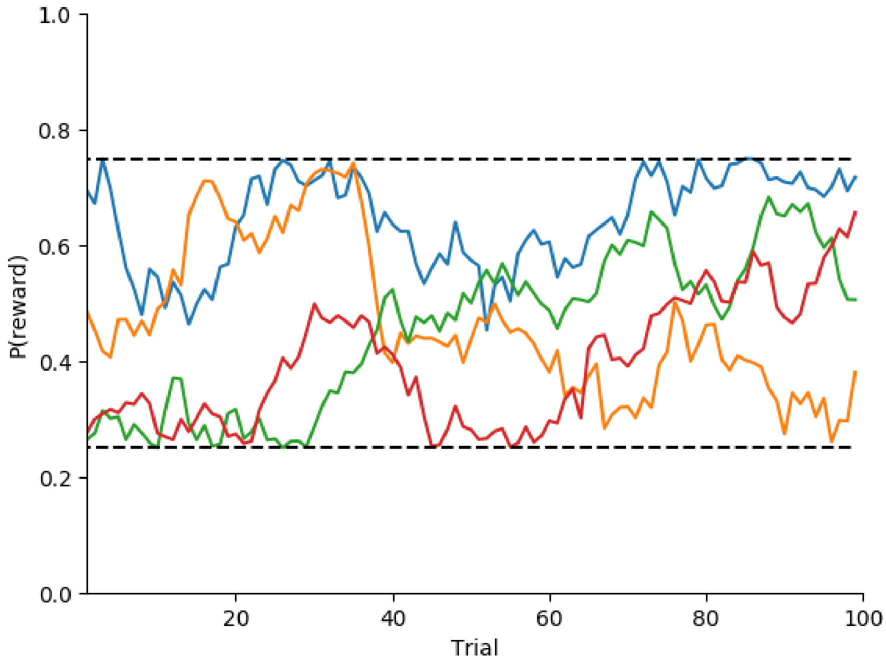


Figure 5.5: Probability of reward at each trial after choosing a second–stage action.

Figure 5.2c demonstrates the prototype behaviours extracted for MF and MB agents. As the figure shows, both prototypes include trials with rare and common transitions and with reward and no-reward. In the case of MB agent, the prototype shows that the agent stays with the same action if the previous trial was rewarded in a common transition, or unrewarded in a rare transition, and otherwise switches to the other action. This is indeed the known signature behaviour of the MB agent in the two-stage task, which was here detected by the model as the group prototype. On the other hand, the MF agent stays with the same action irrespective of whether the transition was common or rare, which is again the signature behaviour of MF agent. Therefore, the method was able to automatically recover signature behaviour of each type of model.

	Log-likelihood			Accuracy %		
Method	Bandit	BD	Two-Stage	Bandit	BD	Two-Stage
GRU	**-0.07**±0.05	-0.60±0.02	**-0.12**±0.01	97.9±1.6	69.6 ±1.0	**96.5**±0.8
Attention	-0.73±0.00	-0.66±0.01	-0.58±0.03	48.0±0.0	69.6±1.0	68.6±19.3
Ablation 1	-0.72±0.06	-0.67±0.03	-0.68±0.02	42.5±28.6	58.0±8.2	58.3±6.5
Ablation 2	0.33±0.02	-0.59±0.08	-0.41±0.06	99.3±0.0	**75.0**±5.5	92.1±5.5
Ablation 3	0.71±0.03	-0.67±0.01	-0.70±0.02	55.1±7.7	61.1±2.0	57.4±2.7
Our	-0.09±0.02	**-0.59**±0.06	-0.29±0.04	**99.6**±0.0	74.2±4.4	93.8±0.8

Table 5.4: Comparison of our method against GRU, Attention Network and the Ablation studies (See Section 5.4.2). The results show the mean log-likelihood and accuracy with one standard deviation across 5 runs. Bolded values indicate the best performance out of the comparators for each dataset and metric.

5.4.2 Ablation studies

To understand which sections of the architecture are critical to classification performance and its interpretation, we analysed the results on each of our datasets when we ablated two different components of the models. The hyperparameters used for each dataset are shown in bold in Table 5.3; and the resulting performance is shown in Table 5.4. The scenarios and results are presented below. We also studied the effect of the length of subsequences, which is presented in Section 5.4.3.

Ablation 1: Performance without subsequence extractor

In order to demonstrate that the model was extracting meaningful and insightful subsequences from the datasets, we trained the model according to only **Stage 1** and **Stage 3**. The subsequence extractor in the network therefore randomly samples subsequences from the original input and attempts to classify those subsequences. Classification performance (Table 5.4) was woeful, suggesting that our model can find subsequences that discriminate between the class much better than random.

Ablation 2: Performance without final linear layer

We investigated the importance of the final linear layer (labelled Stage 3 in Figure 5.1), by removing it from the architecture and retraining the network as according to just **Stage 1** and **Stage 2**. The classification is therefore determined directly by the distance of the prototypes and the embedded subsequence, $\hat{\mathbf{y}}_i = \exp(d_c)/\sum_{c=1}^{C}\exp(d_c)$.

While classification performance without **linear** is similar to that of the original model, Figure 5.2d shows the prototypes produced by ablation experiment for the Two-Stage task. The MF prototype (lower) is essentially the same as that for the full model (shown in Figure 5.2c; lower). However, the MB prototype (upper) no longer presents a recognisable signature. Instead, it exhibits a form of negative reasoning – it is behaviour that the MF agent would *not* exhibit – and so is less interpretable.

Ablation 3: Performance without sequence encoder

While the subsequence encoder, **subenc**, is required in our model to learn the prototypes, we investigated if classification was performed where a (jaccard) similarity metric was computed in the original input space rather than the embedding space produced by **subenc**. The experiment for Ablation 3, went as followed: (1) Train the network as described in Section 5.3 for a given dataset and obtain prototypes for each of the J classes. (2) For a new input sequence, extract the subsequence as determined by the network. (3) Compute the jaccard similarity metric (chosen because features were binary) between the subsequence and the prototypes.. (4) Classify the subsequence based on the highest similarity.

Table 5.4 shows all ablation results. Without the subsequence encoder, the performance of the model performs above random chance suggesting that the subsequences and prototypes extracted by the model are sometimes meaningful for discrimination but does not compare to the performance to the original model. The ablation demonstrates the positive effect of modelling the temporal information of the subsequences using RNNs.

5.4.3 Effect of Subsequence Length, L

We investigate the effect of changing the length of the subsequence, L, being extracted from the input sequence of the performance of our method. We re-optimise the hyperparameter configurations for each L and the results as shown in Figure 5.6. As the length of the subsequence increases, the accuracy improves in the Two Stage and Bandit dataset. Furthermore, when the subsequence length is misspecified, shown in particular by the blue line, where the distinguishing subsequence is designed to be $L = 6$, the variance is large but decreases as L increases.

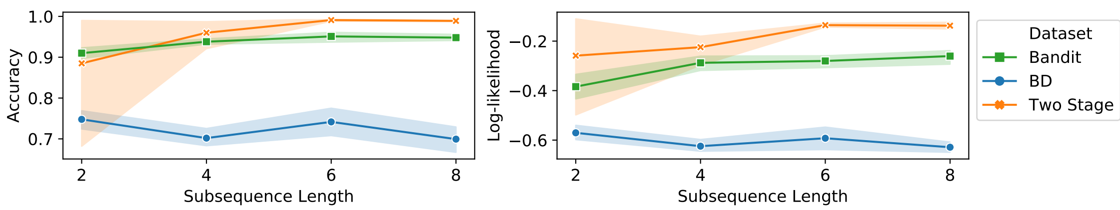


Figure 5.6: The accuracy of classification where the length of the subsequence being extracted and trained by our method is varied.

5.4.4 Quantifying differences between classes

One possible method of quantifying how distinct these prototypes are for each class is to bypass the attentional front-end, and look at the statistics of how well the various prototypes match the embeddings of all subsequences of length L in the input: 1) for a given input sequence, $\mathbf{x_i} = (\mathbf{x}_i^t)_{t=1}^{T_i}$, extract subsequences, from the sequence using a sliding window of length L i.e. we have $T_i - L$ subsequences, $\tilde{\mathbf{x}}_i^k = (\mathbf{x}_i^t)_{t=k}^{k+L-1}$ for $k = 1, 2, ..., T_i - L$, 2) map each subsequence, $\tilde{\mathbf{x}}_i^k$ into the embedding space and calculate the Euclidean distance between the subsequence and each classes' prototype embedding, to produce a vector $\mathbf{d}_i^k \in \mathcal{R}^C$, where C is the number of prototypes, 3) calculate the mean dissimilarity across the subsequences, $\mathbf{d}_i^{mean}$. This is repeated for all other sequences, $\mathbf{x_i}$ for $i = 1, 2, ..., N$ where N is the number of test examples and from that, various summary statistics can be generated. Figure 5.7 shows this for the various datasets, with the means shown for each class and the respective histogram. The result is very clear for synthetic bandit and two-stage task where there is a clear difference between the sequences for each class; progressively less so for BD.

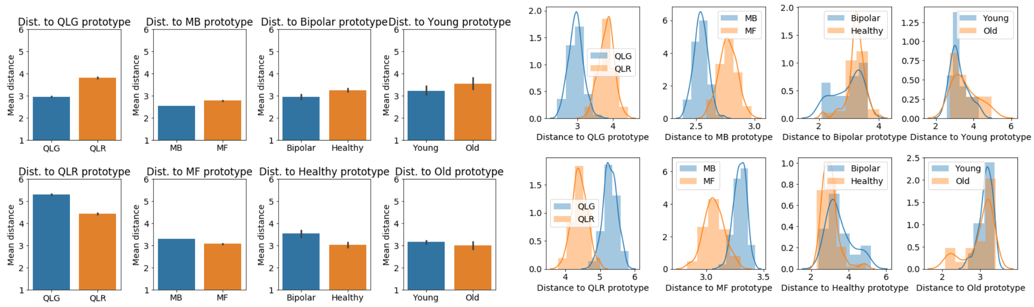


Figure 5.7: Summarizing statistic for the difference between two classes. **Left**: Mean **Right**: Histogram.

5.5 Discussion

Finding signatures of the differences in behaviour between individuals and groups is a critical step towards understanding the nature and idiosyncrasies of the neural and psychological algorithms that generate that behaviour. When powerful, normative, reinforcement learning and Bayesian decision theoretic models fail to describe choices well – for instance, the reward-independent alternation exhibited by bipolar patients in a bandit task – it has historically been left to intuition to guess at what might be different, and hence how the models should be augmented. Here, by exploiting recent ideas in interpretable machine learning [170, 172–174, 178, 179], we have taken an important step towards automating this process – finding prototypical subsequences of actual behaviour that characterise the groups of subjects, and simultaneously determining representative elements of the choices of individuals that underlie their assignment to one of those groups. Our method offers comparable classification performance to that of credible alternatives, whilst affording improved interpretability in both synthetic and clinical datasets. We showed that each element of the architecture plays a critical role.

There are many directions for future work. For instance, just to take the last stage of the network: first, it would be straightforward to use more than one prototype per group, and to expand the dissimilarity matching process accordingly; this could facilitate interpretation of short sequences of behaviour when not all characteristics will be fully on display. Second, we used squared distance to quantify dissimilarity, whereas other metrics might be worth exploring, for instance if signature behaviour in different groups involved characteristically different numbers of episodes in the task. At present, our method could include 'noise' from excess, non-discriminative trials for some classes; a learned metric could help avoid this. Of course, the second stage of the network re-encodes subsequences in light of this squared distance, but this only mitigates some of the concerns.

Another future consideration is the use of transformer networks [174] instead of RNN to process the sequential inputs. The advantage of using transformers is that sequences are processed in its entirety (i.e. the entire trial) rather than action-by-action in this context. However, it could be argued that modelling extremely long-term dependencies is unnecessary as a RNN or even a convolutional network with a small kernel size is more appropriate to model intertemporal human choices. For instance, discounting effects (whether it be hyperbolic or exponential discounting) means that there is an inclination for choices of immediate rewards over rewards that come later into the future [183].

5.6 Summary

In the context of computational psychiatry, there is an impressive range of decision-making tasks that are designed to index sub-components of psychological and neural computations that are distinct across groups of people including people with an underlying disease. To distinguish these differences amongst the groups, current approaches either adopt complex discriminative models—essentially sacrificing interpretability—or use traditional computational models and/or manually-chosen summary statistics at the expense of accuracy and scalability. Especially, applications of these models can be in high-stake situations, there is an inherent need for the decision-making processes associated with machine learning algorithms to be accountable to ensure trust and transparency.

We suggest a method that learns prototypical behaviours of each population in the form of readily interpretable, subsequences of choices, and classifies subjects by finding signatures of these prototypes in their behaviour. The method extends recent suggestions for how the flexibility of recurrent neural networks can be combined with the interpretability of prototypes. The power of the method is illustrated on synthetic and real-world datasets, showing directly that we do not need to sacrifice accuracy for interpretability.

Chapter 6

Conclusion and Future Work

This thesis investigated the application of deep learning models to multiple tasks in neuroscience. Previous Deep neural networks known for their ability to perform complex tasks such as object recognition, machine translation and speech recognition in a multitude of domains. This ability arises from the large number of parameters in the network, allowing to engineer a large number of features at various abstraction levels, while learning the task. Furthermore, these models are able to capture nonlinear relationships within the data which better reflect the nature of the data in several tasks in neuroscience. The methods proposed are applied to several problems in this context such as normalising MRI images into a single domain, translating different modalities of MRI images between each other and modelling the behaviour of humans in sequential decision making tasks.

6.1 Summary of Contribution

The application and development of machine learning methods to several tasks in neuroscience are the contributions presented in this thesis. We propose the use of the CycleGAN for unsupervised normalisation of two distinct MR images from different sites from the same cite in order to improve existing methods that pool data in order to increase sensitivity and statistical power. The thesis further extends the model to be used in semi-supervised learning to translate different modalities of MR images of patients with brain tumours and lesions. Lastly, a deep learning architecture was discussed that models human behaviour on sequential tasks whilst also explaining the characteristics that defines the differences between different groups of participants.

6.1.1 Unsupervised Domain Adaptation for Neuroimaging

Chapter 3 introduces the potential for multi-site studies to present a valuable opportunity to advance research by pooling data in order to increase sensitivity and statistical power. However images derived from MRI are susceptible to both obvious and non-obvious differences between sites which can introduce bias and subject variance, and so reduce statistical power. To rectify these differences, we propose a data driven approach using CycleGANs. Here we transform T1-weighted brain images collected from two different sites into MR images from the same site. The proposed model can reduce site-specific differences without loss of information related to gender (male or female) or clinical diagnosis (schizophrenia or healthy). When trained appropriately, our model is able to normalise imaging sets to a common scanner set with less information loss compared to current approaches.

6.1.2 Semi-Supervised Domain Adaptation using Adversarial Training

While Chapter 3 discusses translating between two distinct MR scanner sets where each patient only has an example image from one scanner, Chapter 4 introduces a novel approach of translating distinct imaging modalities where there is a set of unpaired data and as well as paired data i.e. examples of corresponding images from both modalities. The approach looks particularly at the case where the number of unpaired examples is much larger than the paired examples, and leverages the paired examples to improve the domain translation across multiple modalities. The Semi-Supervised Adversarial CycleGAN uses an adversarial loss to learn from *unpaired* data points, cycle loss to enforce consistent reconstructions of the mappings and another adversarial loss to take advantage of *paired* data points.

In the experiments, the SSA-CGAN is evaluated on multiple datasets on multiple modalities, one with brain tumours, BraTS and another with images of patients with ischaemic lesions, ISLES. The experiments demonstrate that the proposed framework produces an improvement in reconstruction error and reduced variance for the pairwise translation of multiple modalities and is more robust to thermal noise when compared to existing methods.

6.1.3 Interpretable modelling for Neuropsychological Tasks

Chapter 5 introduces problems computational psychiatry and proposes a model that overcomes some of these issues. The aims of this study are to extract insights about the behavioural patterns of each group of subjects and individuals to ultimately predict class label of each subject in an interpretable way. Previous models to achieve these aims either suffer from not being flexible enough to represent a wide range of behaviours resulting in poor classification accuracy, or not being interpretable to extract desired insights.

To address this, here we use the power of recurrent neural networks in producing flexible data representations with the advantages of prototype learning methods for making the representations interpretable. Within an end-to-end classification framework, the method learns a prototype subsequence that is characteristic for each group of subjects and also learns to extract short subsequences from the behaviour of each individual which explain *how* the classification of the individual was made. Due to the discrete nature of subsequences, we use policy gradient to train the models.

Through experiments on synthetic and real world datasets, we show that in terms of classification accuracy the model is comparable to or better than baseline models, while is interpretable and able to extract signature behavioural differences across groups and individuals. The method therefore provides a novel way for interpretable classification of behaviour, which may find applications in different areas such as computational psychiatry.

6.2 Future Research

6.2.1 Development of more efficient models for 3D volumetric data

The major limitation of the method described in this thesis is the restriction to 2D images. Although this loses contextual information provided by the third dimension, this is considered as a form of data augmentation and has proved very successful in tasks such as brain segmentation [184]. The extension to brain volumes could include similar techniques proposed by Wu et al. [185] where convolutions are performed using 3D kernels instead of 2D. MRI volumes can require more memory than can fit on readily available hardware, particularly to train deep networks, and so require more advanced cache management for back propagation. However recent advancements in software such as TensorLy-Torch[1] can be incorporated for future studies.

6.2.2 Including confounds for improved translation

One advantage of conventional regression methods to correct confounds is that they allow for the inclusion of subject-specific covariates such as age and sex. The proposed GAN on the other hand, does not control for covariates and only learns a mapping between scanners while maintaining subject variation. Instead, these co-

[1] http://tensorly.org/torch/dev/

variates must be included as a pre- or post- processing step using standard regression techniques.

Bibliography

(1) A. Payan and G. Montana, "Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks", *arXiv preprint arXiv:1502.02506*, 2015.

(2) S. Sarraf, G. Tofighi et al., "DeepAD: Alzheimer's Disease Classification via Deep Convolutional Neural Networks using MRI and fMRI", *bioRxiv*, 2016, 070441.

(3) K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation", *Medical image analysis*, 2017, **36**, 61–78.

(4) J. Kleesiek, G. Urban, A. Hubert, D. Schwarz, K. Maier-Hein, M. Bendszus and A. Biller, "Deep MRI brain extraction: a 3D convolutional neural network for skull stripping", *NeuroImage*, 2016, **129**, 460–469.

(5) J. H. Cole, R. P. Poudel, D. Tsagkrasoulis, M. W. Caan, C. Steves, T. D. Spector and G. Montana, "Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker", *NeuroImage*, 2017, **163**, 115–124.

(6) R. Marois and J. Ivanoff, "Capacity limits of information processing in the brain", *Trends in Cognitive Sciences*, 2005, **9**, 296–305.

(7) D. World, *Human Brain Facts and Answers*, 2017, https://www.disabled-world.com/health/neurology/brain/bfa.php (visited on 12/21/2020).

(8) C. Australia, *Cancer incidence*, 2015, https://ncci.canceraustralia.gov.au/diagnosis/cancer-incidence/cancer-incidence (visited on 09/20/2020).

(9) C. Australia, *5-year relative survival*, 2015, https://ncci.canceraustralia.gov.au/outcomes/relative-survival-rate/5-year-relative-survival (visited on 09/20/2020).

(10) A. B. of Statistics, *National Survey of Mental Health and Wellbeing: Summary of Results*, 2007, https://www.abs.gov.au/statistics/health/mental-health/national-survey-mental-health-and-wellbeing-summary-results/latest-release (visited on 09/20/2020).

(11) D. of Health Australia, *The magnitude of the problem*, 2009, https://www1.health.gov.au/internet/publications/publishing.nsf/Content/mental-pubs-f-plan09-toc~mental-pubs-f-plan09-con~mental-pubs-f-plan09-con-mag (visited on 12/07/2020).

(12) P. of Australia, *Mental health in Australia: a quick guide*, 2019, https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/rp/rp1819/Quick_Guides/MentalHealth (visited on 12/07/2020).

(13) A. B. of Statistics, *National Health Survey: First results*, 2018, https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/rp/rp1819/Quick_Guides/MentalHealth (visited on 12/07/2020).

(14) S. B. Harvey, M. Deady, M.-J. Wang, A. Mykletun, P. Butterworth, H. Christensen and P. B. Mitchell, "Is the prevalence of mental illness increasing

in Australia? Evidence from national health surveys and administrative data, 2001–2014", *Medical Journal of Australia*, 2017, **206**, 490–493.

(15) A. B. of Statistics, *National Survey of Mental Health and Wellbeing: Summary of Results, 2007*, https://www.abs.gov.au/statistics/health/mental-health/national-survey-mental-health-and-wellbeing-summary-results/latest-release.

(16) M. Institute, *HILDA Survey*, https://melbourneinstitute.unimelb.edu.au/hilda (visited on 09/20/2020).

(17) D. C. Lam, P. M. Salkovskis and H. M. Warwick, "An experimental investigation of the impact of biological versus psychological explanations of the cause of "mental illness"", *Journal of Mental Health*, 2005, **14**, 453–464.

(18) A. Dezfouli, K. Griffiths, F. Ramos, P. Dayan and B. W. Balleine, "Models that learn how humans learn: the case of decision-making and its disorders", *PLoS computational biology*, 2019, **15**, e1006903.

(19) P. Das, A. H. Kemp, G. Flynn, A. W. Harris, B. J. Liddell, T. J. Whitford, A. Peduto, E. Gordon and L. M. Williams, "Functional disconnections in the direct and indirect amygdala pathways for fear processing in schizophrenia", *Schizophrenia research*, 2007, **90**, 284–294.

(20) X. Zhan and R. Yu, "A window into the brain: advances in psychiatric fMRI", *BioMed research international*, 2015, **2015**.

(21) R. C. Petersen, P. Aisen, L. A. Beckett, M. Donohue, A. Gamst, D. J. Harvey, C. Jack, W. Jagust, L. Shaw, A. Toga et al., "Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization", *Neurology*, 2010, **74**, 201–209.

(22) B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest et al., "The multimodal brain

tumor image segmentation benchmark (BRATS)", *IEEE Transactions on Medical Imaging*, 2014, **34**, 1993–2024.

(23) O. Maier, B. H. Menze, J. von der Gablentz, L. Häni, M. P. Heinrich, M. Liebrand, S. Winzeck, A. Basit, P. Bentley, L. Chen et al., "ISLES 2015-A public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI", *Medical image analysis*, 2017, **35**, 250–269.

(24) R. A. Poldrack and K. J. Gorgolewski, "OpenfMRI: Open sharing of task fMRI data", *Neuroimage*, 2017, **144**, 259–261.

(25) Y. Ding, J. H. Sohn, M. G. Kawczynski, H. Trivedi, R. Harnish, N. W. Jenkins, D. Lituiev, T. P. Copeland, M. S. Aboian, C. Mari Aparici et al., "A deep learning model to predict a diagnosis of Alzheimer disease by using 18F-FDG PET of the brain", *Radiology*, 2019, **290**, 456–464.

(26) E. Hosseini-Asl, R. Keynton and A. El-Baz, "Alzheimer's disease diagnostics by adaptation of 3D convolutional network", *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, 126–130.

(27) R. J. Meszlényi, K. Buza and Z. Vidnyánszky, "Resting state fMRI functional connectivity-based classification using a convolutional neural network architecture", *Frontiers in Neuroinformatics*, 2017, **11**, 61.

(28) D.-S. Huang, K. Han and M. Gromiha, *Intelligent Computing in Bioinformatics*, Springer, 2014.

(29) M. C. Tjepkema-Cloostermans, R. C. de Carvalho and M. J. van Putten, "Deep learning for detection of focal epileptiform discharges from scalp EEG recordings", *Clinical Neurophysiology*, 2018, **129**, 2191–2196.

(30) R. A. Poldrack and K. J. Gorgolewski, "Making big data open: data sharing in neuroimaging", *Nature Neuroscience*, 2014, **17**, 1510–1517.

(31) F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning", *arXiv preprint arXiv:1702.08608*, 2017.

(32) M. T. Ribeiro, S. Singh and C. Guestrin, “” Why should I trust you?” Explaining the predictions of any classifier”, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, 1135–1144.

(33) I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial nets”, *Advances in Neural Information Processing Systems*, 2014, 2672–2680.

(34) B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest et al., “The multimodal brain tumor image segmentation benchmark (BRATS)”, *IEEE Transactions on Medical Imaging*, 2015, **34**, 1993–2024.

(35) J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks”, *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2223–2232.

(36) C. M. Bishop, *Pattern recognition and machine learning*, springer, 2006.

(37) A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, *Communications of the ACM*, 2017, **60**, 84–90.

(38) K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, 1724–1734.

(39) A. Graves, A.-r. Mohamed and G. Hinton, “Speech recognition with deep recurrent neural networks”, *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, 6645–6649.

(40) A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 3128–3137.

(41) J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", *arXiv preprint arXiv:1412.3555*, 2014.

(42) D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", *Nature*, 1986, **323**, 533–536.

(43) Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, 1994, **5**, 157–166.

(44) X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, 249–256.

(45) T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude", *COURSERA: Neural networks for machine learning*, 2012, **4**, 26–31.

(46) D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, ed. Y. Bengio and Y. LeCun.

(47) X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks", *Proceedings of the 14th international Conference on Artificial Intelligence and Statistics*, 2011, 315–323.

(48) S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, 1997, **9**, 1735–1780.

(49) J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 248–255.

(50) A. Krizhevsky, *Learning multiple layers of features from tiny images*, tech. rep., 2009.

(51) C. R. Jack Jr, M. A. Bernstein, N. C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. J. Britson, J. L. Whitwell, C. Ward et al., "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods", *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 2008, **27**, 685–691.

(52) J. E. Moody, "Note on generalization, regularization and architecture selection in nonlinear learning systems", *Neural Networks for Signal Processing Proceedings of the 1991 IEEE Workshop*, 1991, 1–10.

(53) G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", *CoRR*, 2012.

(54) N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *The Journal of Machine Learning Research*, 2014, **15**, 1929–1958.

(55) C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization", *Neural Computation*, 1995, **7**, 108–116.

(56) L. Prechelt, "Early stopping-but when?", *Neural Networks: Tricks of the trade*, 1998, 55–69.

(57) C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of Big Data*, 2019, **6**, 1–48.

(58) S. Wold, K. Esbensen and P. Geladi, "Principal component analysis", *Chemometrics and intelligent laboratory systems*, 1987, **2**, 37–52.

(59) L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE”, *Journal of Machine Learning Research*, 2008, **9**, 2579–2605.

(60) G. E. Hinton, “Deep belief networks”, *Scholarpedia*, 2009, **4**, 5947.

(61) D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes”, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

(62) Y. Bengio, E. Laufer, G. Alain and J. Yosinski, “Deep generative stochastic networks trainable by backprop”, *International Conference on Machine Learning*, 2014, 226–234.

(63) I. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks”, *arXiv preprint arXiv:1701.00160*, 2016.

(64) X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang and S. Paul Smolley, “Least squares generative adversarial networks”, *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2794–2802.

(65) S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele and H. Lee, “Generative adversarial text to image synthesis”, *International Conference on Machine Learning*, 2016, 1060–1069.

(66) T. Karras, T. Aila, S. Laine and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation”, *arXiv preprint arXiv:1710.10196*, 2017.

(67) Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu and Z. Fang, “Towards the automatic anime characters creation with generative adversarial networks”, *arXiv preprint arXiv:1708.05509*, 2017.

(68) C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang et al., “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.”, *CVPR*, 2017, **2**, 4.

(69) P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-image translation with conditional adversarial networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 1125–1134.

(70) O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015, 234–241.

(71) T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz and B. Catanzaro, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8798–8807.

(72) Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim and J. Choo, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8789–8797.

(73) J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang and E. Shechtman, Advances in neural information processing systems, 2017, pp. 465–476.

(74) A. B. L. Larsen, S. K. Sønderby, H. Larochelle and O. Winther, International conference on machine learning, 2016, pp. 1558–1566.

(75) J. Donahue, P. Krähenbühl and T. Darrell, "Adversarial feature learning", *arXiv preprint arXiv:1605.09782*, 2016.

(76) V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky and A. Courville, "Adversarially learned inference", *arXiv preprint arXiv:1606.00704*, 2016.

(77) R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

(78) R. S. Sutton, "Learning to predict by the methods of temporal differences", *Machine learning*, 1988, **3**, 9–44.

(79) C. J. C. H. Watkins, "Learning from delayed rewards", 1989.

(80) W. H. Organisation, *Mental disorders*, 2021.

(81) A. P. Association et al., *Diagnostic and statistical manual of mental disorders (DSM-5®)*, American Psychiatric Pub, 2013.

(82) A. T. Beck and B. A. Alford, *Depression: Causes and treatment*, University of Pennsylvania Press, 2009.

(83) H. Grunze, "Bipolar disorder", *Neurobiology of Brain Disorders*, 2015, 655–673.

(84) B. Müller-Oerlinghausen, A. Berghöfer and M. Bauer, "Bipolar disorder", *The Lancet*, 2002, **359**, 241–247.

(85) T. M. Laursen, M. Nordentoft and P. B. Mortensen, "Excess early mortality in schizophrenia", *Annual Review of Clinical Psychology*, 2014, **10**, 425–448.

(86) T. R. Insel, "Rethinking schizophrenia", *Nature*, 2010, **468**, 187–193.

(87) D. Weishaupt, V. D. Köchli and B. Marincek, *How does MRI work?: an introduction to the physics and function of magnetic resonance imaging*, Springer Science & Business Media, 2008.

(88) R.-J. M. Van Geuns, P. A. Wielopolski, H. G. de Bruin, B. J. Rensing, P. M. van Ooijen, M. Hulshoff, M. Oudkerk and P. J. de Feyter, "Basic principles of magnetic resonance imaging", *Progress in Cardiovascular Diseases*, 1999, **42**, 149–156.

(89) M. A. Bernstein, K. F. King and X. J. Zhou, *Handbook of MRI pulse sequences*, Elsevier, 2004.

(90) D. G. Mitchell and M. S. Cohen, "MRI principles", 2004.

(91) R. D. Fields, "White matter matters", *Scientific American*, 2008, **298**, 54–61.

(92) J. J. Sheldon, R. Siddharthan, J. Tobias, W. A. Sheremata, K. Soila and M. Viamonte, "MR imaging of multiple sclerosis: comparison with clinical and CT examinations in 74 patients", *American journal of neuroradiology*, 1985, **6**, 683–690.

(93) L. de Boer, J. Axelsson, K. Riklund, L. Nyberg, P. Dayan, L. Bäckman and M. Guitart-Masip, "Attenuation of dopamine-modulated prefrontal value signals underlies probabilistic reward learning deficits in old age", *Elife*, 2017, **6**, e26424.

(94) N. D. Daw, S. J. Gershman, B. Seymour, P. Dayan and R. J. Dolan, "Model-based influences on humans' choices and striatal prediction errors", *Neuron*, 2011, **69**, 1204–1215.

(95) G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, University of Cambridge, Department of Engineering Cambridge, UK, 1994, vol. 37.

(96) C. M. Gillan, M. Kosinski, R. Whelan, E. A. Phelps and N. D. Daw, "Characterizing a psychiatric symptom dimension related to deficits in goal-directed control", *Elife*, 2016, **5**, e11305.

(97) C. M. Gillan, A. R. Otto, E. A. Phelps and N. D. Daw, "Model-based learning protects against forming habits", *Cognitive, Affective, & Behavioral Neuroscience*, 2015, **15**, 523–536.

(98) A. Rao, J. M. Monteiro, J. Mourao-Miranda, A. D. Initiative et al., "Predictive modelling using neuroimaging data in the presence of confounds", *NeuroImage*, 2017, **150**, 23–49.

(99) A. Abdulkadir, O. Ronneberger, S. J. Tabrizi and S. Klöppel, "Reduction of confounding effects with voxel-wise Gaussian process regression in structural MRI", *2014 International Workshop on Pattern Recognition in Neuroimaging*, 2014, 1–4.

(100) J. Dukart, M. L. Schroeter, K. Mueller, A. D. N. Initiative et al., "Age correction in dementia–matching to a healthy brain", *PloS One*, 2011, **6**, e22193.

(101) D. Kostro, A. Abdulkadir, A. Durr, R. Roos, B. R. Leavitt, H. Johnson, D. Cash, S. J. Tabrizi, R. I. Scahill, O. Ronneberger et al., "Correction of inter-scanner and within-subject variance in structural MRI based automated diagnosing", *NeuroImage*, 2014, **98**, 405–415.

(102) H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function", *Journal of Statistical Planning and Inference*, 2000, **90**, 227–244.

(103) T. Hergueta, R. Baker and G. C. Dunbar, "The Mini-International Neuropsychiatric Interview (MINI): the development and validation of a structured diagnostic psychiatric interview for DSM-IVand ICD-10", *J clin psychiatry*, 1998, **59**, 2233.

(104) M. B. First, R. L. Spitzer, M. Gibbon, J. B. Williams et al., *Structured clinical interview for DSM-IV-TR axis I disorders, research version, patient edition*, tech. rep., SCID-I/P, 2002.

(105) J. Ashburner, "A fast diffeomorphic image registration algorithm", *Neuroimage*, 2007, **38**, 95–113.

(106) J. Ashburner and K. J. Friston, "Computing average shaped tissue probability templates", *Neuroimage*, 2009, **45**, 333–341.

(107) M. A. Yassa and C. E. Stark, "A quantitative evaluation of cross-participant registration techniques for MRI studies of the medial temporal lobe", *Neuroimage*, 2009, **44**, 319–327.

(108) J. Ashburner and K. J. Friston, "Voxel-based morphometry—the methods", *Neuroimage*, 2000, **11**, 805–821.

(109) C. D. Good, I. Johnsrude, J. Ashburner, R. N. Henson, K. J. Friston and R. S. Frackowiak, "Cerebral asymmetry and the effects of sex and handedness on brain structure: a voxel-based morphometric analysis of 465 normal adult human brains", *Neuroimage*, 2001, **14**, 685–700.

(110) A. Mechelli, K. J. Friston, R. S. Frackowiak and C. J. Price, "Structural covariance in the human cortex", *Journal of Neuroscience*, 2005, **25**, 8303–8310.

(111) T. Kim, M. Cha, H. Kim, J. K. Lee and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks", *International Conference on Machine Learning*, 2017, 1857–1865.

(112) X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang and S. Paul Smolley, "Least squares generative adversarial networks", *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2794–2802.

(113) S. Nowozin, B. Cseke and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization", *Advances in Neural Information Processing Systems*, 2016, 271–279.

(114) M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein generative adversarial networks", *International Conference on Machine Learning*, 2017, 214–223.

(115) Y. Mroueh, T. Sercu and V. Goel, "Mcgan: Mean and covariance feature matching gan", *International Conference on Machine Learning*, 2017, 2527–2535.

(116) D. Ulyanov, A. Vedaldi and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization", *arXiv preprint arXiv:1607.08022*, 2016.

(117) B. Magnin, L. Mesrob, S. Kinkingnéhun, M. Pélégrini-Issac, O. Colliot, M. Sarazin, B. Dubois, S. Lehéricy and H. Benali, "Support vector machine-

based classification of Alzheimer's disease from whole-brain anatomical MRI", *Neuroradiology*, 2009, **51**, 73–83.

(118) C. Jongkreangkrai, Y. Vichianin, C. Tocharoenchai, H. Arimura, A. D. N. Initiative et al., "Computer-aided classification of Alzheimer's disease based on support vector machine with combination of cerebral image features in MRI", *Journal of Physics: Conference Series*, 2016, **694**, 012036.

(119) J. L. Winterburn, A. N. Voineskos, G. A. Devenyi, E. Plitman, C. de la Fuente-Sandoval, N. Bhagwat, A. Graff-Guerrero, J. Knight and M. M. Chakravarty, "Can we accurately classify schizophrenia patients from healthy controls using magnetic resonance imaging and machine learning? A multi-method and multi-dataset study", *Schizophrenia Research*, 2017.

(120) C. Davatzikos, D. Shen, R. C. Gur, X. Wu, D. Liu, Y. Fan, P. Hughett, B. I. Turetsky and R. E. Gur, "Whole-brain morphometric study of schizophrenia revealing a spatially complex set of focal abnormalities", *Archives of General Psychiatry*, 2005, **62**, 1218–1227.

(121) N. Koutsouleris, E. M. Meisenzahl, C. Davatzikos, R. Bottlender, T. Frodl, J. Scheuerecker, G. Schmitt, T. Zetzsche, P. Decker, M. Reiser et al., "Use of neuroanatomical pattern classification to identify subjects in at-risk mental states of psychosis and predict disease transition", *Archives of General Psychiatry*, 2009, **66**, 700–712.

(122) T. Zhang, N. Koutsouleris, E. Meisenzahl and C. Davatzikos, "Heterogeneity of structural brain changes in subtypes of schizophrenia revealed using magnetic resonance imaging pattern analysis", *Schizophrenia Bulletin*, 2014, **41**, 74–84.

(123) J. Kambeitz, L. Kambeitz-Ilankovic, S. Leucht, S. Wood, C. Davatzikos, B. Malchow, P. Falkai and N. Koutsouleris, "Detecting neuroimaging biomark-

ers for schizophrenia: a meta-analysis of multivariate pattern recognition studies", *Neuropsychopharmacology*, 2015, **40**, 1742.

(124) H. G. Schnack and R. S. Kahn, "Detecting neuroimaging biomarkers for psychiatric disorders: sample size matters", *Frontiers in Psychiatry*, 2016, **7**, 50.

(125) J. Mourao-Miranda, A. L. Bokde, C. Born, H. Hampel and M. Stetter, "Classifying brain states and determining the discriminating activation patterns: support vector machine on functional MRI data", *NeuroImage*, 2005, **28**, 980–995.

(126) J. Mourao-Miranda, A. Reinders, V. Rocha-Rego, J. Lappin, J. Rondina, C. Morgan, K. D. Morgan, P. Fearon, P. B. Jones, G. A. Doody et al., "Individualized prediction of illness course at the first psychotic episode: a support vector machine MRI study", *Psychological Medicine*, 2012, **42**, 1037–1047.

(127) A. N. Ruigrok, G. Salimi-Khorshidi, M.-C. Lai, S. Baron-Cohen, M. V. Lombardo, R. J. Tait and J. Suckling, "A meta-analysis of sex differences in human brain structure", *Neuroscience & Biobehavioral Reviews*, 2014, **39**, 34–50.

(128) S. M. Plis, D. R. Hjelm, R. Salakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. J. Johnson, J. S. Paulsen, J. A. Turner and V. D. Calhoun, "Deep learning for neuroimaging: a validation study", *Frontiers in Neuroscience*, 2014, **8**.

(129) G. L. Wenk et al., "Neuropathologic changes in Alzheimer's disease", *Journal of Clinical Psychiatry*, 2003, **64**, 7–10.

(130) D. Lu, K. Popuri, G. W. Ding, R. Balachandar and M. F. Beg, "Multimodal and multiscale deep neural networks for the early diagnosis of Alzheimer's

disease using structural MR and FDG-PET images", *Scientific reports*, 2018, **8**, 5697.

(131) M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin and H. Larochelle, "Brain tumor segmentation with deep neural networks", *Medical Image Analysis*, 2017, **35**, 18–31.

(132) A. Mahbod, M. Chowdhury, Ö. Smedby and C. Wang, "Automatic brain segmentation using artificial neural networks with shape context", *Pattern Recognition Letters*, 2018, **101**, 74–79.

(133) D. Dai, J. Wang, J. Hua and H. He, "Classification of ADHD children through multimodal magnetic resonance imaging", *Frontiers in Systems Neuroscience*, 2012, **6**, 63.

(134) J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation", *arXiv preprint arXiv:1711.03213*, 2017.

(135) E. Tzeng, J. Hoffman, K. Saenko and T. Darrell, "Adversarial discriminative domain adaptation", *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, 2017, 7167–7176.

(136) M. B. McDermott, T. Yan, T. Naumann, N. Hunt, H. Suresh, P. Szolovits and M. Ghassemi, "Semi-Supervised Biomedical Translation with Cycle Wasserstein Regression GANs", *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

(137) E. L. Denton, S. Chintala, R. Fergus et al., "Deep generative image models using a laplacian pyramid of adversarial networks", *Advances in Neural Information Processing Systems*, 2015, 1486–1494.

(138) A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", *arXiv preprint arXiv:1511.06434*, 2015.

(139) X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks", *European Conference on Computer Vision*, 2016, 318–335.

(140) Z. Yi, H. (Zhang, P. Tan and M. Gong, "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation.", *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2868–2876.

(141) M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks", *Advances in Neural Information Processing Systems*, 2016, 469–477.

(142) Y. Aytar, L. Castrejon, C. Vondrick, H. Pirsiavash and A. Torralba, "Cross-modal scene networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, **40**, 2303–2314.

(143) T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, "Improved techniques for training gans", *Advances in Neural Information Processing Systems*, 2016, 2234–2242.

(144) T. Miyato, S.-i. Maeda, S. Ishii and M. Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

(145) S. Park, J. Park, S.-J. Shin and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning", *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

(146) A. Beers, J. Brown, K. Chang, J. P. Campbell, S. Ostmo, M. F. Chiang and J. Kalpathy-Cramer, "High-resolution medical image synthesis using progressively grown generative adversarial networks", *arXiv preprint arXiv:1805.03144*, 2018.

(147) P. Seeböck, S. M. Waldstein, S. Klimscha, H. Bogunovic, T. Schlegl, B. S. Gerendas, R. Donner, U. Schmidt-Erfurth and G. Langs, "Unsupervised

Identification of Disease Marker Candidates in Retinal OCT Imaging Data", *IEEE Transactions on Medical Imaging*, 2018.

(148) H. Yang, J. Sun, A. Carass, C. Zhao, J. Lee, Z. Xu and J. Prince, "Unpaired Brain MR-to-CT Synthesis Using a Structure-Constrained CycleGAN", *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2018, 174–182.

(149) C. Chen, Q. Dou, H. Chen, J. Qin and P.-A. Heng, "Synergistic Image and Feature Adaptation: Towards Cross-Modality Domain Adaptation for Medical Image Segmentation", *arXiv preprint arXiv:1901.08211*, 2019.

(150) C. S. Perone, P. Ballester, R. C. Barros and J. Cohen-Adad, "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling", *NeuroImage*, 2019, **194**, 1–11.

(151) P. Welander, S. Karlsson and A. Eklund, "Generative adversarial networks for image-to-image translation on multi-contrast MR images-A comparison of CycleGAN and UNIT", *arXiv preprint arXiv:1806.07777*, 2018.

(152) M.-Y. Liu, T. Breuel and J. Kautz, "Unsupervised image-to-image translation networks", *Advances in Neural Information Processing Systems*, 2017, 700–708.

(153) N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich and J. C. Gee, "N4ITK: improved N3 bias correction", *IEEE Transactions on Medical Imaging*, 2010, **29**, 1310.

(154) W. H. Organization, *Cause-specific mortality, estimates for 2000–2012*, 2012, http://www.who.int/healthinfo/global_burden_disease/estimates/en/index1.%20html (visited on 12/21/2020).

(155) J. Johnson, A. Alahi and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution", *European Conference on Computer Vision*, 2016, 694–711.

(156) K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 770–778.

(157) M. Mathieu, C. Couprie and Y. LeCun, "Deep multi-scale video prediction beyond mean square error", *arXiv preprint arXiv:1511.05440*, 2015.

(158) P. Faulkner, Q. Huys, D. Renz, N. Eshel, S. Pilling, P. Dayan and J. Roiser, "A Comparison of Pruning During Multi-Step Planning in Depressed and Healthy Individuals", *bioRxiv*, 2020.

(159) P. Garety, D. Hemsley and S. Wessely, "Reasoning in deluded schizophrenic and paranoid patients: biases in performance on a probabilistic inference task.", *Journal of Nervous and Mental Disease*, 1991.

(160) P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)", *A Practical Guide, 1st Ed.*, 2017.

(161) N. D. Daw et al., "Trial-by-trial data analysis using computational models", *Decision making, affect, and learning: Attention and performance XXIII*, 2011, **23**.

(162) A. Dezfouli, H. Ashtiani, O. Ghattas, R. Nock, P. Dayan and C. S. Ong, "Disentangled behavioural representations", *Advances in Neural Information Processing Systems*, 2019, 2251–2260.

(163) B. Lau and P. W. Glimcher, "Dynamic response-by-response models of matching behavior in rhesus monkeys", *Journal of the Experimental Analysis of Behavior*, 2005, **84**, 555–579.

(164) A. Dezfouli, R. Morris, F. Ramos, P. Dayan and B. W. Balleine, "Integrated accounts of behavioral and neuroimaging data using flexible recurrent neural network models", *bioRxiv*, 2018, 328849.

(165) A. Karpathy, J. Johnson and L. Fei-Fei, "Visualizing and understanding recurrent networks", *arXiv preprint arXiv:1506.02078*, 2015.

(166) H. Strobelt, S. Gehrmann, H. Pfister and A. M. Rush, "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks", *IEEE Transactions on Visualization and Computer Graphics*, 2017, **24**, 667–676.

(167) D. Alikaniotis, H. Yannakoudakis and M. Rei, "Automatic Text Scoring Using Neural Networks", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, 715–725.

(168) W. J. Murdoch, P. J. Liu and B. Yu, "Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs", *International Conference on Learning Representations*, 2018.

(169) W. J. Murdoch and A. Szlam, "Automatic rule extraction from long short term memory networks", *arXiv preprint arXiv:1702.02540*, 2017.

(170) Z. Che, S. Purushotham, R. Khemani and Y. Liu, "Interpretable deep models for ICU outcome prediction", *AMIA Annual Symposium Proceedings*, 2016, **2016**, 371.

(171) C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead", *Nature Machine Intelligence*, 2019, **1**, 206–215.

(172) D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.

(173) M.-T. Luong, H. Pham and C. D. Manning, "Effective approaches to attention-based neural machine translation", *arXiv preprint arXiv:1508.04025*, 2015.

(174) A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need", *Advances in Neural Information Processing Systems*, 2017, 5998–6008.

(175) K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention", *International Conference on Machine Learning*, 2015, 2048–2057.

(176) E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism", *Advances in Neural Information Processing Systems*, 2016, 3504–3512.

(177) O. Li, H. Liu, C. Chen and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions", *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

(178) C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin and J. K. Su, "This looks like that: deep learning for interpretable image recognition", *Advances in Neural Information Processing Systems*, 2019, 8928–8939.

(179) Y. Ming, P. Xu, H. Qu and L. Ren, "Interpretable and steerable sequence learning via prototypes", *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, 903–913.

(180) K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, 1724–1734.

(181) R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning", *Machine learning*, 1992, **8**, 229–256.

(182) T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor", *International Conference on Machine Learning*, 2018, 1861–1870.

(183) S. Frederick, G. Loewenstein and T. O'donoghue, "Time discounting and time preference: A critical review", *Journal of economic literature*, 2002, **40**, 351–401.

(184) S. González-Villà, A. Oliver, S. Valverde, L. Wang, R. Zwiggelaar and X. Lladó, "A review on brain structures segmentation in magnetic resonance imaging", *Artificial Intelligence in Medicine*, 2016, **73**, 45–69.

(185) J. Wu, C. Zhang, T. Xue, B. Freeman and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling", *Advances in Neural Information Processing Systems*, 2016, 82–90.