DISS. ETH NO. 27007

Towards MRI data analysis via learning on graphs

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES OF ETH ZURICH
(DR. SC. ETH ZURICH)

presented by

MARIA ELISABETTA GHISU
MASTER OF SCIENCE ETH IN MATHEMATICS

BORN ON 21 SEPTEMBER 1989
CITIZEN OF ITALY

accepted on the recommendation of

Prof. Dr. Karsten Borgwardt
Prof. Dr. Tanja Stadler
PD. Dr. Gabriele Lohmann

2020

ABSTRACT

In the last 50 years, technological advancements have shaped the progress in numerous areas, from medicine and biology, to social and political sciences. Machine learning models have started to extend in the clinical domain, to enable personalized treatments and precision medicine tasks. The goal of these algorithms is to support experts' diagnosis and facilitate early detection and prevention of diseases, while tracking their progression. Thus, the creation of complete and organised health databases, to store patients records and medical history, is crucial to the success of automated predictive algorithms. Additionally, the growing abundance of data generation coming in different formats, from hospital discharge to imaging and genetic samples, require the development of new systems designed to handle the data structure and problem of interest.

Amongst all, medical imaging data have received considerable interest lately. Their collection is consistently integrating as part of standard clinical routines, partially due to more accurate scanner protocols and economically affordable machines. In particular, brain Magnetic Resonance Imaging (MRI) data are invaluable resources to unfold the understanding of this complex organ, whose anatomy and functionality are still highly mysterious. From an analytical perspective, brain MRI data give rise to multiple challenges. For instance, the discrepancy across samples, caused by intrinsic noise, or differences of the scanner and individual brain anatomy, requires consistent and robust processing to guarantee a reliable evaluation. Moreover, the multitude of biomarkers and feature extraction strategies, culminates in a variety of data modalities and structures, all carrying complementary information form the same input.

Navigating the heterogeneity of data and methodologies in brain MRI is far from being trivial. In this thesis, we address these challenges from different perspectives and integrate methodologies for the analysis of graphs, to eventually model diverse structured biomedical data.

In the first part, we perform a multi-modal and multi-task analysis on two cohorts of brain imaging data, including patients diagnosed with Multiple Sclerosis and Depression, as well as healthy individuals. We extract whole-brain features, either from high-resolution voxels or aggregating summary statistics from regions of interest. We compare several classification strategies and establish an optimal pipeline for health status prediction. Furthermore, we investigate the complex tasks of treatment response, disease progression, and patients subtype categorization. We propose to use multiple kernel learning methods to combine information from different modalities, and evaluate their impact on the prediction. Our findings are variegated. Depression health status and response to Electroconvulsive therapy (ECT) of patients can be predicted with significant accuracy, while resulting in clinically interpretable brain activation patterns.

Complex tasks, including unsupervised patient subtypes and assessment of disease progression, still require more investigation. In our cohort, combining images from multiple modalities can improve over their individual counterpart on selected scenarios, although we often observe the performance to be dominated by the single best modality.

In the second part of this thesis, we propose novel methodologies for the analysis of graph structured data, with applications on molecular property prediction tasks. We develop a similarity measure on graphs inspired by optimal assignment solutions, calculating a distance over the distribution of their node embeddings. Our approach evaluates the difference between substructures by computing local similarities, overcoming the limitation of classical aggregation steps. We extend the successful Weisfeiler–Lehman propagation scheme to graphs with continuous attributes, and outperform the state-of-the-art classification performance in several benchmark data sets. Subsequently, we introduce a framework to extend transfer learning on graph structured data, by enhancing Graph Neural Network (GNN) models with adversarial layers. Employing shared knowledge from large molecular data sets to small target specific domains, we improve the prediction on multiple experimental settings.

We conclude our work envisioning the next steps of our research, and detailing the ideas to integrate the individual contributions of this thesis. Undoubtedly, graphs are flexible structures to encode different data types, while brain MRI have a natural graph representation given by their anatomical and functional connection. Besides, the low sample availability is known as a major shortcoming in the clinical data domain, and particularly in imaging studies. From this perspective, it is undeniable that transfer learning will play a crucial role in the years to come, to guarantee efficient extension of successful machine learning models in the field of healthcare.

SOMMARIO

Negli ultimi 50 anni, gli avanzamenti della tecnologia hanno modellato il progresso in numerose aree dalla medicina e la biologia, alle scienze sociali e politiche. I modelli dell'apprendimento automatico hanno iniziato ad estendersi nell'ambito clinico, per permettere trattamenti personalizzati e medicina di precisione. Lo scopo di questi algoritmi è quello di supportare gli esperti nelle diagnosi e di facilitare la diagnosi precoce e la prevenzione di malattie, allo stesso tempo tracciando il loro progresso. Ne consegue che la creazione di banche dati sanitarie complete e organizzate, per contenere la cartella clinica del paziente e la sua storia medica, è fondamentale per il successo di algoritmi automatici predittivi. Inoltre, la crescente abbondanza nella generazione di dati in diversi formati, dai moduli di dimissione, alle immagini e ai campioni genetici, richiede lo sviluppo di nuovi sistemi per gestire le strutture dati e il problema in questione.

Tra tutti, i dati delle immagini mediche hanno recentemente suscitato un notevole interesse. Il loro raccoglimento si sta constantemente integrando negli esami di routine, anche grazie a protocolli di scanner piu accurati e macchine a prezzi accesibili. In particolare, i dati rilevati dalle immagini a risonanza magnetica (Magnetic Resonance Imaging; MRI) del cervello, sono risorse inestimabili per svelare e comprendere la struttura di questo organo complesso, la cui anatomia e funzionalità sono ancora un mistero. Da una prospettiva analitica, i dati MRI del cervello presentano diverse sfide. Per esempio, la discrepanza tra i campioni, a causa del rumore di fondo, o differenze negli scanner e nell'anatomia individuale del cervello, rechiede approcci solidi per garantire una valutazione affidabile. Inoltre, l'esistente moltitudine di biomarcatori e strategie per l'estrazione di caratteristiche, culmina in una varietà di strutture e modalità di dati, ognuno contenente informazioni complementari dalla stessa sorgente.

Navigare l'eterogeneità di dati e metodologie per gli MRI del cervello è tutt'altro che banale. In questa tesi, affrontiamo queste sfide da diverse prospettive e integriamo metodologie per l'analisi di grafi, con lo scopo finale di modellare una diversità di dati biomedici strutturati.

Nella prima parte, eseguiamo un'analisi multi-modale e multi-tasking in due gruppi di dati di immagini del cervello, che includono pazienti affetti da depressione e sclerosi multipla, così come individui sani. Estraiamo caratteristiche dall'intero cervello, sia tramite voxels ad alta risoluzione, sia dall'aggregazione di statistiche ottenute da regioni di interesse. Compariamo diversi classificatori e stabiliamo una pipeline ottimale per la predizione dello stato di salute. Inoltre, investighiamo i complessi problemi della risposta al trattamento, della progressione delle malattie, e della categorizzazione dei sottogruppi di pazienti. Proponiamo di usare metodi kernel multipli per combinare l'informazione tra diverse modalità di dati, valutandone l'impatto sul problema di

predizione. Le nostre scoperte sono variegate. Lo stato di salute della depressione e la risposta alla terapia elettroconvulsiva (Electroconvulsive therapy; ECT) dei pazienti possono essere predetti con un'accuratezza significativa, risultando in pattern di attivazione del cervello clinicamente interpretabili. Problemi complessi, che includono l'apprendimento non supervisionato dei sottogruppi di pazienti e la valutazione della progressione della malattia, rechiedono maggiore investigazione. Nel nostro gruppo, la combinazione di immagini in diverse modalità puo migliorare sulla singola controparte in scenari selezionati, sebbene osserviamo spesso che la prestazione è dominata dalla singola migliore modalità di immagine.

Nella seconda parte di questa tesi, proponiamo una nuova metodologia per l'analisi di strutture dati grafo, con applicazioni nella predizione di proprietà delle molecole. Sviluppiamo una misura di similarità tra grafi ispirata da soluzioni per i problemi di assegnazione, calcolando una distanza tra distribuzioni di embedding di nodi. Il nostro approccio valuta la differenza tra sotto-strutture, computando similarità locali, superando le limitazioni dei classici step di aggregazione. Estendiamo lo schema di propagazione Weisfeiler–Lehman, che ha gia avuto molto successo, a grafi con attributi continui, e superiamo le prestazioni di classificazione dello stato dell'arte in numerosi set di dati di riferimento. Successivamente, introduciamo un framework per estendere il transfer learning su strutture dati grafo, migliorando i modelli di Graph Neural Networks (GNNs) con livelli antagonisti (adversarial layers). Condividendo la conoscenza da un grande campione di dati molecolari a piccoli domini specifici (target domains), miglioriamo la predizione in molti protocolli sperimentali.

Concludiamo questo lavoro prefigurando gli step successivi della nostra ricerca ed esponendo le idee per integrare le singole contribuzioni di questa tesi. Indubbiamente, i grafi sono strutture flessibili che possono codificare diversi tipi di dati, mentre gli MRI del cervello posseggono una naturale rappresentazione a grafo generata dalle loro connessioni anatomiche e funzionali. Inoltre, la scarsa disponibilità di campioni di dati è nota per essere uno dei maggiori difetti nell'ambito dei dati clinici, specialmente negli studi con le immagini. In questa prospettiva, è innegabile che il transfer learning giocherà un ruolo cruciale negli anni avvenire, per garantire l'efficiente estensione dei metodi di successo dell'apprendimento automatico nel campo della sanità.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Dr. Karsten Borgwardt for the excellent guidance and advice during the years of my PhD, both at the professional and personal level. I am particularly grateful for the freedom he gave me to pursue my own research path, never lacking to provide constant support, encouragement, and inspiration.

I would like to thank Prof. Dr. Tanja Stadler and PD. Dr. Gabriele Lohmann for kindly agreeing to be part of my doctoral examination committee, as external referees. I also thank Kobi Benenson for chairing the examination.

I am deeply thankful to all my co-authors, without them a lot of the work in this thesis would have not been possible. As a main contributor in a lot of projects, I cannot emphasize enough how great it was to work with Matteo Togninalli. He has been a constant source of motivation and inspiration, as well as an excellent colleague with who I shared plenty of interesting discussions and ideas. I am so glad that I also had the chance to know him as friend and office mate, and for the many funny, honest, and "complaining" moments.

Matti Gärtner has been an amazing collaborator throughout the years. I would like to thank him for the fruitful Skype discussions and for his invaluable contribution on the clinical interpretability of our findings. I also thank Simone Grimm, Francesco Grussu, Antonio Ricciardi, Claudia Wheeler–Kingshott, and Anke Henning for their insights on the MRI studies.

I am deeply grateful to Felipe Llinares-López for being a great mentor, source of endless knowledge and advise. I always appreciated his direct opinions on both the technical and political aspects of my PhD, and he certainly shaped the kind of researcher I am today.

I would like to sincerely thank Catherine Jutzeler for her valuable insights on the writing of this thesis. I also thank Max Horn for his comments on parts of this manuscript, and Bastian Rieck for providing the LaTeX template.

My time during the PhD would have not been as great without the current and past members of the MLCB Lab. I am so happy I could share an office with Anja Gumpinger and Max Horn, thank you for the many fun and work related chats, and for always making my daily hours more pleasant. Thank you to Bastian Rieck for being a great collaborator in many of my projects, and for the multiple scientific and philosophical discussions. A special thanks goes to Damián Roqueiro, I highly appreciated his constant support and availability, on the research, software, and personal side. I also thank Dean Bodenham, for his kindness and advices in the early years of my PhD. Thank you to Laetitia Papaxanthos, Udo Gieraths, Menno Witteveen, Dominik Grimm, Xiao He, Thomas Gumbsch, Caroline Weis, Lukas Folkmann, Christian Bock,

Michael Moor, Daisuke Yoneoka, Catherine Jutzeler, Leslie O'Bray, Juliane Klatt, and Lucie Bourguignon for the engaging technical discussions, conference experiences, and fun after work time together. I would especially like to thank Giulia Muzio, for keeping some "Italianity" within the lab and for being to me a nice reminder of the beginning. Finally, I would like to thank Katharina Heinrich and Cindy Malnasi for their precious help in all the administrative matters.

These years in Zurich, Basel, and at the D-BSSE have been positively shaped by the great people I have met. A special thank goes to Olivier, Arthur, Masha, and Selen, for plenty of drinks, coffee, and chats. I would like to thank Silvia, for the many runs and dinners, and for being my Italian connection in Basel. Thank you to Clara and Luca for being a certainty in my Zurich life, and for their great insights on this mysterious world outside academia. Thank you to Arianna for the fun flat sharing time. Thank you to Giulia, because as we shared different phases of this Swiss and PhD adventure, I could always be the real me with her.

I am so lucky to have found so many awesome friends in my life. My Gaeta's friends are the people I have known since I was born. Thank you all for still making the summer a time to look forward to, and special thanks to Edoardo, Federico, Martina, Chicca, Matteo, Gaetano, and Tommaso. Rome is always a good place to come back, because in the end, there is where we still reunite. Amongst all, I would like to especially thank Luca, Giacomo, Simone, Enzo, and Valentina for the great time that we keep on sharing after all these years.

I will never be thankful enough to Alessia and Manuela, because they have always seen the best in me. Without them, my academic career would have probably not even started.

The best thing about Martina, is that I do not have to explain why I am thankful to her, because she already knows. If I had to summarise, I would simply say thank you for being my anchor in every possible aspect of my life.

I would like to thank my parents, Francesco and Maria Teresa, for their support and for always leaving me the freedom to follow my own path. I also thank my aunt and uncle, Chiara and Luciano, for making every family dinner more enjoyable. I thank my grandmother Alba Maria, because I am confident that she has been the only person who always believed in me. Finally, I would like to thank my brother Gualtiero, for inspiring me every day to become a better version of myself.

Contents

I An overview on graphs and MRI via transfer learning 1

1 Introduction 3
- 1.1 Motivation . . . 3
- 1.2 Brain MRI for studying neurological disorders . . . 4
- 1.3 Graph modelling of clinical data . . . 6
- 1.4 Transfer learning . . . 7
- 1.5 Organisation and contributions of this thesis . . . 8
  - 1.5.1 Kernels and neural networks for graph structured data . . . 8
  - 1.5.2 Classification of depression health status with brain MRI . . . 9
  - 1.5.3 Analysing complex neurological tasks . . . 9
  - 1.5.4 Wasserstein Weisfeiler-Lehman kernel . . . 10
  - 1.5.5 Adversarial Graph Neural Networks . . . 11
  - 1.5.6 Outlook and appendix . . . 11

2 From kernels to neural networks for graph structured data 13
- 2.1 An overview of kernel methods . . . 14
  - 2.1.1 Reproducing Kernel Hilbert Spaces . . . 14
  - 2.1.2 Kernels . . . 15
  - 2.1.3 $\mathcal{R}$-convolution framework . . . 18
- 2.2 Graph kernels . . . 18
  - 2.2.1 Preliminaries on graphs . . . 19
  - 2.2.2 Graph kernels based on nodes or edges . . . 21
  - 2.2.3 Graph kernels based on walks and paths . . . 23
  - 2.2.4 Graph kernels based on sub-graphs . . . 24
  - 2.2.5 Graph kernels based on iterative label refinement . . . 25
  - 2.2.6 Beyond the $\mathcal{R}$–convolution framework . . . 27
- 2.3 Graph Neural Networks . . . 28
  - 2.3.1 The Graph Neural Network model . . . 29
  - 2.3.2 The graph isomorphism problem in GNNs . . . 30

II Multi-modal multi-task analysis of neurological and psychiatric disorders 31

3 Classification of patients and healthy individuals using brain MRI 33
3.1 Data description . . . 34
3.1.1 Major depressive disorder study . . . 34
3.1.2 Multiple sclerosis study . . . 37
3.2 Pattern analysis methods in neuroimaging . . . 39
3.2.1 Univariate analysis: statistical parametric map . . . 39
3.2.2 Multivariate classification analysis for brain imaging data . . . 41
3.2.3 Multi-modal analysis . . . 42
3.3 Feature extraction . . . 46
3.3.1 High-dimensional features . . . 47
3.3.2 Low resolution region of interest features . . . 48
3.4 Experiments . . . 49
3.4.1 Experimental setup . . . 50
3.4.2 Results . . . 53
3.5 Discussion . . . 59

4 Predicting complex tasks 63
4.1 Treatment response prediction in depression . . . 64
4.1.1 Data and feature extraction . . . 64
4.1.2 Methods . . . 65
4.1.3 Experiments . . . 68
4.2 Identifying patient subtypes in Multiple Sclerosis . . . 71
4.2.1 Methods and results . . . 71
4.3 Discussion . . . 73
4.3.1 Treatment response . . . 73
4.3.2 Complex tasks in Multiple Sclerosis . . . 74

III Learning on graphs 77

5 Wasserstein Weisfeiler-Lehman graph kernels 79
5.1 Optimal transport . . . 80
5.1.1 Wasserstein distance . . . 81
5.2 Wasserstein distance on graphs . . . 84
5.2.1 Graph embedding scheme . . . 84
5.2.2 Graph Wasserstein Distance . . . 87
5.3 From distance to kernels . . . 89
5.3.1 Definiteness of the WWL . . . 90
5.3.2 Kreĭn Support Vector Machines . . . 96
5.4 Experiments . . . 97
5.4.1 Data sets . . . 97

5.4.2 Experimental setup . . . 98
5.4.3 Classification results . . . 99
5.4.4 Runtime and complexity . . . 101
5.5 Discussion . . . 103

6 Adversarial graph neural networks 105
6.1 Transfer learning . . . 106
6.1.1 Supervised transfer learning . . . 107
6.1.2 Multi-task learning . . . 108
6.1.3 The scenario of domain adaptation . . . 108
6.1.4 Domain adversarial training . . . 109
6.2 Adversarial layers for graph neural networks . . . 109
6.2.1 Adversarial graph neural networks . . . 110
6.2.2 Supervised transfer learning . . . 112
6.2.3 Multi-task adversarial learning . . . 113
6.2.4 Pre-training graph neural networks . . . 114
6.3 Experiments . . . 115
6.3.1 Data sets . . . 115
6.3.2 Experimental setup . . . 116
6.3.3 Results on molecular datasets . . . 118
6.3.4 Pre-training adversarial GNN . . . 119
6.4 Discussion . . . 120

IV Summary and Outlook 123

7 Conclusions and outlook 125
7.1 Further exploration of neurological tasks . . . 125
7.2 Extending Wasserstein kernels . . . 127
7.3 Perspectives in domain adaptation . . . 128
7.4 A unified framework for graphs in brain MRI . . . 129
7.4.1 Understanding domain adaptation for MRI . . . 131
7.5 Conclusion . . . 132

A Magnetic resonance imaging: modalities, acquisition, preprocessing, and analysis 135
A.1 Magnetic resonance imaging . . . 135
A.1.1 Structural MRI . . . 135
A.1.2 Functional MRI . . . 136
A.2 Major Depressive Disorder study . . . 136
A.2.1 Data acquisition and preprocessing . . . 136
A.2.2 Additional results . . . 137
A.3 Multiple Sclerosis study . . . 137
A.3.1 Data acquisition . . . 137

B SOFTWARE AVAILABILITY 141
B.1 A clinical decision system for MRI data: software integration 141
B.1.1 Data upload 141
B.1.2 Feature extraction 141
B.1.3 Model training 142
B.1.4 System integration 142
B.2 A software framework to compute graph kernels 142
B.2.1 How to use `graphkernels` 143

BIBLIOGRAPHY 145

List of Figures

1.1 MRI acquisition and segmentation 5
1.3 Brain graph 7

3.1 Multiple sclerosis disease subtypes 37
3.3 Multiple Kernel Learning pipeline on the MDD data 46
3.4 Machine learning pipeline on task-based fMRI for the MDD study 48
3.5 ROI parcellation with GIF 49
3.6 Performance of multiple beta images with a linear SVM. 53
3.7 Performance of multiple beta images with an rbfSVM 54
3.8 Classifiers comparison on the neutral stimuli condition 55
3.9 Average SVM weights in ROIs 57
3.10 SVM weight map 57
3.11 Precision-Recall curves for uni-modal and multi-modal classifiers 60
3.13 Precision-Recall curves with Easy Multiple Kernel Learning 61

4.1 Machine learning pipeline for the classification and regression analysis . 67
4.2 Classification weight maps on sMRI data 69
4.3 Descriptive statistics of clinical predictors in ECT responders and non-responders 70
4.4 Regression results on aPHCr region 71
4.5 PCA for the ROI features 73

5.1 Optimal mass transportation problem 81
5.2 Illustration of the optimal transport problem on the bakery example . . 82
5.3 Schematic view of the discrete optimal transport problem 83
5.4 Weisfeiler–Lehman embedding scheme 86
5.5 Wasserstein distance 87
5.6 Runtime performance of the WWL 103

6.1 Schematic view of adversarial Graph Neural Networks 112
6.2 Schematic view of task-shared adversarial GNN 113

A.1 Performance on Beta images with a polynomial kernel of degree 2. . . . 138
A.2 Performance on Beta images with a polynomial kernel of degree 3. . . . 139
A.3 Performance on Beta images with a sigmoid kernel. 139

List of Tables

3.1 Descriptive statistics of the MDD study cohort 35
3.2 Beta and contrast images 47
3.4 Classification results on contrast images 56
3.5 Classification results of the multi-modal analysis 58

5.1 Description of the experimental data sets. 97
5.2 Classification accuracies on categorical graphs 99
5.3 Classification accuracies on continuously attributed graphs 100
5.4 Classification accuracies on synthetic graphs. 101

6.1 Data set information 115
6.2 Classification results in the unsupervised domain adaptation scenario . 118
6.3 Classification results in the supervised transfer learning scenario 119
6.4 Classification results with pre-training 120

A.1 Classification results on contrast images 138

Part I

An overview on graphs and MRI via transfer learning

1 INTRODUCTION

> *"It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to change."*
>
> Leon C. Megginson – previously attributed to Charles Darwin

The path leading to the writing of this thesis has gone throughout a multitude of unforeseen swings and fluctuations over the years of my PhD. The appealing idea of integrating recent deep learning developments with graph based models, to detect relevant patterns and features from brain images, has met the inevitable reality of data limitations. To overcome the issue of missing information, researchers came up with the elegant solution of transfer learning. This technique permits knowledge translation across domains and exploits the abundance of different sources from related problems, to solve the tricky ones. In this work, we investigate through all these aspects, starting with classical analysis of MRI data, followed by the development of a novel graph kernel, and concluding with the expansion of domain adaptation strategies on graph structured data. We envision the integration of our advances in graph learning with MRI data analysis to be the ultimate outlook of this dissertation, progressing the ambitious and fascinating goal of understanding the human brain.

1.1 MOTIVATION

Of all the organ systems in the human body, the brain holds the record for being the most complex and challenging when it comes to unveiling its processes. It serves as an intermediary from the external stimuli of the outside world to our perception of them, besides controlling our movements, speech and thoughts. Nevertheless, our biological understanding of the brain, its functionality and anatomy, is far from being complete. Medical advances over the last 50 years have made enormous progresses towards a better understanding of this complex organ, with genetic and imaging data playing a crucial role for it. While medical data acquisition is increasingly becoming an integrated part of the standard clinical routine, the harmonized and standardized data collection as well as data curation remain major challenges. In general, the collected features and samples will not be comparable across studies posing a major challenge for the data analysis and interpretation. Furthermore, imaging techniques are still bound to a high economical cost, with the consequence that physicians must carefully select which data

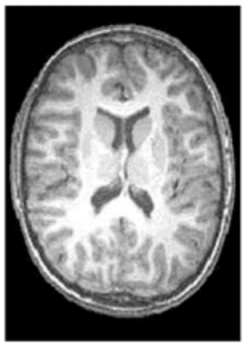
modalities to collect, leading to possible biases. As opposed to the tech related fields that are hallmarked by an exponential explosion of data availability (e.g., smartphones, smartwatches, social media, and web services), this growth is moving at a slower pace in the medical domain. Noisy and small data possibly represent the primary obstacle for the application and innovation of machine learning methods. This is particularly pronounced in the deep learning area, where neural networks have reached top level accuracy and exceeded the ability of human experts [106, 163], but still require a large data set for model training. This problem has not been ignored by researchers, with transfer learning being proposed as the most promising approach to surpass the shortcomings arising from limited domain knowledge [139, 201]. The success of the transfer learning idea, to improve model learning capabilities by transferring information across domains, has been supported by its positive impact on different research areas [32, 117, 146]. In addition, meaningful representation of medical data is far from being trivial, due to the variety of features and their hidden interactions. From images throughout time series, to graphs and tabular data, users are presented with multiple choices for visualizing and structuring their data. By virtue of their structural flexibility, graphs are a powerful tool to represent different types of objects and their connections, from patients and diseases in knowledge graphs, towards brain structures in medical images, to chemical bonds and genetic interaction networks. This thesis bridges the gap across these research areas. Learning to transfer information across structured data, offers the opportunity to extend machine learning models on different applications, where intrinsic data limitations have been an obstacle in the past.

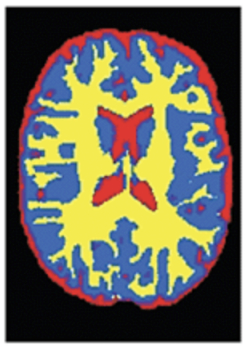
1.2 Brain MRI for studying neurological disorders

The anatomy of the brain is per se conglomerated: a standard segmentation divides it into White Matter (WM), Grey Matter (GM), and cerebrospinal fluid (CSF). WM tracts, made of nerve fibres (axons), transmit impulses between neurons which constitute the GM, ultimately forming the structural connection across brain areas. Besides the anatomical structure, a multitude of activations and signals occur during static times and as a reaction to external stimuli. These activations generate the so-called functional connectivity of the brain. A detailed picture of the brain can be obtained by means of magnetic resonance imaging (MRI). This is a non invasive technique that outputs high resolution images using a magnetic field, generated while a subject lies inside a scanner machine (Figure 1.1). Depending on the information of interest, different modalities of MRI can be acquired; at an high-level, they separate between structural and functional, accordingly to the type of connectivity detected. Figure 1.1 shows a typical acquisition protocol in the clinic with a Siemens scanner (left panel), and an example of the outcome, a structural MRI image, with standard segmentation into WM, GM, and CSF (right panel). While certain alterations of functional and structural connectivity are part of the natural evolution, unexpected changes are typically the consequence, or early signs, of neurological and psychiatric medical conditions. MRI analysis is particularly useful for studying these disorders, in the context of diagnosis, as well as early detec-

Figure 1.1: A standard MRI acquisition protocol (left) with a Siemens scanner in the clinic. The outcome is an MRI structural image, with segmentation into white matter (yellow), grey matter (blue), and cerebrospinal fluid (red).

(a) Siemens scanner.[a]





(b) Example of a T1 segmentation. Source: Nagel and Kroenke [128].

[a]Source https://www.indiamart.com/proddetail/siemens-trio-3t-mri-scanner-20917637562.html

tion and treatment response prediction. Neurological disorders affect and damage the nervous system, with the most common including Parkinson, Alzheimer, Schizophrenia and Multiple Sclerosis. Other diseases causing mental illness, such as Major and Bipolar depression, are categorized as psychiatric disorders. Nevertheless, as depression directly affects the brain connectivity it has been discussed whether to include it in the neurological group [95]. Within this thesis, we will refer to neurological disorders also including depression. Either mental and psychiatric diseases, as well as neurological conditions, have been widely investigated with the help of MRI images. Providing a complete picture of the brain they constitute a major step forward to understand and study such diseases.

Affecting more than 264 million people worldwide, clinical depression is among the most common mental disorders [93]. Depression is related to disruption in the cognitive domain, specifically affecting working memory related tasks [188]. Neuroimaging studies have tried to highlight the dysfunctional areas and mechanisms affected in the brain, but showed conflicting results. To solve these inconsistencies, replicating previous studies on a larger cohort is a first and important step. Additionally, employing multivariate machine learning analysis rather than classical statistical inference [55], helps to circumvent the multiple comparison problem [46] while detecting distributed patterns of activity.

Similar considerations apply for other neurological disorder, such as Multiple Sclerosis (MS), a neurodegenerative disease affecting approximately 2 million people worldwide [91]. As MS damages the white matter tracts, MRI images are valuable instruments to detect these injuries and enable early diagnosis. Nonetheless, the unpredictable course of MS is still a major limitation for the diagnosis, since subjects with similar symptoms might evolve to totally different severity levels over the years. Moreover,

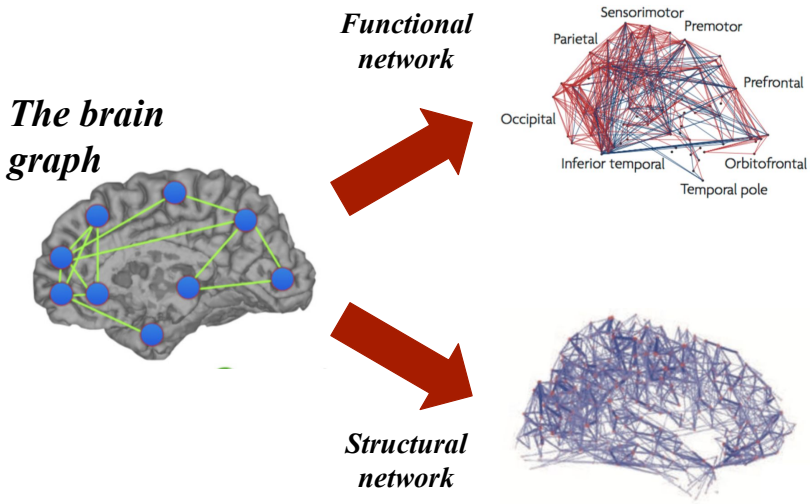
MRI images often report lesions undetectable by the human eye. Machine learning has helped to find and interpret the subtle aberrations in the MRI, to support prediction of early diagnosis of MS and other complex neurological tasks [173].

Classical analysis of MRI data extracts image based features as input for the learning algorithm. Relevant information can be obtained either at whole-brain or regional level. In the latter case, patches are selected containing knowledge and characteristics associated to individual diseases, which are known to affect different brain areas. An alternative to the standard image feature representation of MRI data is tu use graph structures. Thanks to their flexibility, graphs permit to unveil and efficiently combine hidden patterns from the images and can also be exploited to integrate MRI data with other type of clinical information.

1.3 Graph modelling of clinical data

A lot of the knowledge coming from real world domains, from social networks, smartphone and mobility data, through signal processing and software, to healthcare and genetic data, cannot be represented in basic vectorial representation but require complex data structures. Graphs are used to represent relations between objects, and are then extremely useful for many of these applications. In biology, networks can represent protein-protein interaction or genes interaction, with the edges depicting functional relationships. In a more general context, knowledge graphs encode any kind of relational information between different sources. For example, in healthcare, we can construct a knowledge graph by relating subjects with diseases, hospitals or symptoms. In chemoinformatics, graphs have been widely used to model compounds. For a molecule, atoms correspond to nodes while their bond represent the edges. Finally, graphs can encode the complex structure of the brain, either showing structural or functional connectivities [27, 164]. At the functional level, the nodes represent brain areas, while edges form the correlation between their activity; at the structural level, the network is defined by the anatomical connection between regions or brain tissue. A schematic overview of the brain graph is provided in Figure 1.3. Approaches have been proposed to analyse and compare brain graphs, mostly based on the extraction of topological properties. Modern machine learning techniques can directly take the graphs as input for the learning algorithms, exploiting nodes and edge features then employing various propagation schemes along the graph to extract vectorial representations. These methods are mainly divided into two categories, kernel based and deep learning approaches, primarily differing in the procedure to determine the weights of the model. Overall, the advancements of network analysis strategies, and in particular the exploitation of complex graph substructures, has lead to improved performance in many fields, when compared to standard approaches. As a consequence, many researchers have focused on the development of graph based analysis techniques for brain MRI studies, pursuing the ambitious goal of bringing the understanding of the brain and its disorders to a new level. Overall, despite the numerous data collection efforts, research in the MRI area still suffers from the lack of homogeneous and large

Figure 1.3: A representation of the brain graph for functional and structural networks. Adapted from multiple sources: Islam *et al.* [92] and Heuvel *et al.* [83]



cohorts. To address this problem, the efficient machine learning approach is to apply transfer learning.

1.4 Transfer learning

Transfer learning has been developed with the goal to improve model learning, when difficulties arise from the data itself, due to small sample size or incomplete domain knowledge. In a nutshell, the idea is to learn a model exploiting information from a source domain and adapt it to the limited target domain. In image analysis, these methods have shown to be effective, where it is well known that low-level features capture general properties, while deeper features are task specific [201]. In general, the transfer can occur at different levels, including model parameters or task and input related knowledge. A key question is to understand how and what to transfer in order to guarantee sufficient similarity across the source and target, and avoid negative transfer, which hurts the model performance [152]. To date, a profound understanding of transferability still remains an open problem. In healthcare applications, and in particular on medical imaging, this research is still at the dawn, due to the difficulty to find good source data to transfer from. A recent work by Raghu *et al.* [146] performed a comprehensive evaluation studying the effect of transfer learning on model performance, from the ImageNet [42] database to various medical imaging tasks, including the diagnosis of diabetic retinopathy and five different diseases from chest x-rays. The authors report the gain offered from transfer learning to be negligible, with small models reaching comparable performances. Exploring the learned features, they also observed that transferred models tend to overfit, suggesting that hybrid approaches which only adapt part of the network are the most promising for future investigation. According to this study,

understanding how to maximize the gain from learning large scale models in medical imaging domains, is still an open and active research area. Similar considerations apply on the brain MRI domain, where transfer learning studies are even rarer and more problematic, due to the intrinsic data inconsistencies across different studies.

The successive step of integrating transfer learning with graph based algorithms is far from being trivial. A major issue is due to the lack of a straightforward feature interpretation where the data structure is heterogeneous across samples, as it occurs for graphs with different topology. Recent work attempted to combine transfer learning with various graph neural network models, taking advantage from the deep learning perspective, while the application areas spaced from chemoinformatics to text classification [89, 194]. Nevertheless, efficient domain adaptation on brain MRI graphs is yet an unexplored topic. We hope that the growing interest and development in graph transfer learning methods could also motivate the neuroimaging community to further develop these techniques, overcoming the long lasting problem of data limitation.

1.5 Organisation and contributions of this thesis

In this section we detail the main contributions of this thesis and present the organisation of the text. We will provide a brief summary of each of the upcoming chapters, listing the of corresponding publications as well as individual contributions. The content of this thesis is presented in four parts:

(i) An introduction and background, with an overview of kernels and neural network approaches for graph structured data;

(ii) Analysis of multi-modal and multi-task brain MRI data, with application to studies on major depression and multiple sclerosis;

(iii) Learning on graph structured data, from the development of a new kernel to transfer learning on graphs;

(iv) Conclusion and outlook for future work.

Part of the Introduction (Chapter 1), is based on all the publications listed below.

1.5.1 Kernels and neural networks for graph structured data

In Chapter 2 we present a short introduction to kernel methods, followed by a an overview of the current state-of-the-art in graph kernels. We categorize graph kernels according to the type of substructure and aggregation strategy used to build the similarity matrix, from the most popular $\mathcal{R}$ – convolution framework to recently developed optimal assignment approaches. Then, we offer a brief self-contained description of graph neural networks, with particular attention to the message passing framework, and highlight their connection to the class of Weisfeiler–Lehman kernels. Part of this chapter is based on the following review:

- Karsten Borgwardt, **Elisabetta Ghisu**, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. Graph Kernels. State-of-the-art and future challenges. *Foundations and Trends in Machine Learning*, 2020.

  Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck structured the review and designed the experiments. Bastian Rieck performed the experiments, with contributions from Elisabetta Ghisu, Felipe Llinares-López, and Leslie O'Bray. All authors wrote the manuscript. The list of authors is ordered alphabetically.

1.5.2 Classification of depression health status with brain MRI

The collection of brain MRI data has exploded in the clinic in the latest years, with scanner machines becoming more accurate and cheaper, providing an invaluable tool for medical doctors to detect early signs of diseases. With a variety of available imaging modalities, it is unclear which type should be collected for the task of interest. In Chapter 3 we study the patients versus control classification task, in a cohort of healthy individuals and patients diagnosed with depression using task-based fMRI data and employing multivariate analysis tools. We find that subjects can be successfully separated into the two clinical groups, based on different evaluation metrics. Inferring the most informative features from the classification model, we detect relevant patterns in the brain that are associated with either patients or controls. We then experiment on the integration of different MRI modalities with multiple kernel learning techniques, and discuss their ultimate contribution to improve the learning performance. Part of this chapter is based on the following publication:

- Matti Gärtner*, **Elisabetta Ghisu***, Milan Scheidegger, Luisa Bönke, Yan Fan, Anna Stippl, Ana-Lucia Herrera-Melendez, Sophie Metz, Emilia Winnebeck, Marial Fissler, Anke Henning, Malek Bajbouj, Karsten Borgwardt, Thorsten Barnhofer, and Simone Grimm. Aberrant working memory processing in major depression: evidence from multivoxel pattern classification. *Neuropsychopharmacology 43, no. 9 (2018): 1972–1979.* * = Equal contribution.

  Matti Gärtner, Elisabetta Ghisu, Karsten Borgwardt, and Simon Grimm designed the study. Matti Gärtner performed the pre-processing and post-hoc region of interest analysis. Elisabetta Ghisu performed the machine learning classification analysis. Matti Gärtner and Simone Grimm contributed to the clinical interpretation of results, with support from Elisabetta Ghisu on the analytical side. Matti Gärtner and Elisabetta Ghisu wrote the manuscript, with contributions from Karsten Borgwardt, Simone Grimm, and all other authors.

1.5.3 Analysing complex neurological tasks

While detecting the patients versus control phenotype is one of the most common problem in biological and medical applications, it is far more interesting and relevant for the clinic to study complex tasks, such as early detection of diseases or treatment response.

We investigate this problem in Chapter 4, where we analyse MRI samples from patients diagnosed with major depression and MS, testing supervised and unsupervised machine learning techniques to predict individual therapy responses and identify patient subtypes. We further discuss the limitations and challenges of our study, with an outlook on the open problem of predicting the progression of neurological disorders. Part of this chapter is based on the following manuscript:

- Matti Gärtner, **Elisabetta Ghisu**, Ana Lucia Herrera-Mendelez, Michael Koslowski, Sabine Aust, Patrick Asbach, Christian Otte, Francesca Regen, Isabella Heuser, Karsten Borgwardt, Simone Grimm*, Malek Bajbouj*. Using routine MRI data of depressed patients to predict individual responses to electroconvulsive therapy. *Experimental neurology (2020): 113505*. * = Equal contribution.

  Matti Gärtner, Elisabetta Ghisu, Simon Grimm, and Malek Bajbouj designed the study. Matti Gärtner processed the data and performed the clinical analysis. Elisabetta Ghisu performed the machine learning experiments, including the classification and regression analysis. Matti Gärtner, Simon Grimm, and Malek Bajbouj contributed to the clinical interpretation of results, with support from Elisabetta Ghisu on the analytical side. Matti Gärtner and Elisabetta Ghisu wrote the manuscript, Karsten Borgwardt, Simone Grimm, Malek Bajbouj, and all other authors.

1.5.4 Wasserstein Weisfeiler-Lehman kernel

In Chapter 5 we present one of the main methodological contributions of this thesis, the Wasserstein–Weisfeiler Lehman Kernel (WWL). To create a new similarity measure, we represent graphs as distributions of node embeddings and utilize tools from optimal transport theory to evaluate their distance. We develop a theoretical framework to support the validity of our approach, extending the Wasserstein Distance on graph structured data. We evaluate the performance in terms of runtime and accuracy, comparing WWL with other state-of-the-art graph kernels on real-world and synthetic data sets. This chapter is based on the following publication:

- Matteo Togninalli*, **Elisabetta Ghisu***, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. Wasserstein Weisfeiler-Lehman graph kernels. *In Advances in Neural Information Processing Systems, pp. 6439-6449*. 2019. * = Equal contribution.

  Matteo Togninalli, Elisabetta Ghisu, Bastian Rieck, and Karsten Borgwardt conceived the research. Matteo Togninalli and Elisabetta Ghisu performed the experiments, with contributions from Bastian Rieck. Matteo Togninalli and Bastian Rieck proved the theoretical results, with contributions from Elisabetta Ghisu. Matteo Togninalli and Elisabetta Ghisu wrote the paper, with contributions from Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt.

1.5.5 Adversarial Graph Neural Networks

As the neuralized version of kernel methods, graph neural networks have gained a lot of attention in the latest years, although they are usually confined to large sample size regimes. In Chapter 6 we propose a general approach to apply transfer learning on graph neural networks, improving their learning capabilities on limited sample size domains. We present a comprehensive empirical evaluation and conclude with a critical discussion to establish the conditions for effective adversarial based domain adaptation on graphs. This chapter is based on the following manuscript:

- **Elisabetta Ghisu**, Matteo Togninalli, Felipe Llinares-López, and Karsten Borgwardt. Adversarial Graph Neural Networks. *In submission.*

  Elisabetta Ghisu, Matteo Togninalli, Felipe Llinares-López, and Karsten Borgwardt conceived the project. Elisabetta Ghisu performed the experiments, with contributions from Matteo Togninalli. Elisabetta Ghisu wrote the paper, with contributions from Matteo Togninalli, Felipe Llinares-López, and Karsten Borgwardt.

1.5.6 Outlook and appendix

We conclude this work by detailing ideas for future directions to explore within our research. In particular, we present an outline to integrate the graph based methodology developed in this thesis with the clinical application on the brain MRI domain. In the appendix, we describe the software created for the different sub-projects. Part of the content discussed in Chapter 7 and Appendix B is based on the following published work:

- Christian Bock*, Matteo Togninalli*, **Elisabetta Ghisu**, Thomas Gumbsch, Bastian Rieck, and Karsten Borgwardt. A Wasserstein Subsequence Kernel for Time Series. *In 2019 IEEE International Conference on Data Mining (ICDM), pp. 964-969*. IEEE, 2019. * = Equal contribution

  Christian Bock, Matteo Togninalli, Elisabetta Ghisu, Thomas Gumbsch, Bastian Rieck, and Karsten Borgwardt conceived the research. Christian Bock and Matteo Togninalli performed the experiments, with contributions from Elisabetta Ghisu and Thomas Gumbsch. Elisabetta Ghisu implemented the kernel baselines. Christian Bock, Matteo Togninalli, and Bastian Rieck wrote the paper, with contributions from Elisabetta Ghisu, Thomas Gumbsch and Karsten Borgwardt.

- Mahito Sugiyama, **Elisabetta Ghisu**, Felipe Llinares-López, and Karsten Borgwardt. graphkernels: R and Python packages for graph comparison. *Bioinformatics 34, no. 3 (2018) 530–532*.

  Mahito Sugiyama, Elisabetta Ghisu, Felipe Llinares-López, and Karsten Borgwardt designed the work. Mahito Sugiyama coded the backend `C/C++` interface and the `R` library. Elisabetta Ghisu implemented the frontend `Python` package and wrapper from `C/C++`. Mahito Sugiyama and Elisabetta Ghisu wrote the application note, with contributions from Felipe Llinares-López and Karsten Borgwardt.

2 From kernels to neural networks for graph structured data

Kernel methods have been extensively developed in the last decades to provide an expressive representation of many real world data, to uncover hidden relations and patterns, and facilitate the applicability of learning algorithms. Classical machine learning methods for regression, classification, or clustering are designed to take as input a tabular data matrix, where each row is a sample and the columns represent categorical or continuous features. On one hand, in the setting of high dimensional data, i.e. when the number of features is very large, using this *explicit* feature representation becomes computationally infeasible. On the other hand, many real word data such as images, graphs, and time series are challenging for standard machine learning algorithms, because they do not come in a vectorial representation. For example, a 2D picture has a spatial component that cannot be retrieved by simply flattening the pixel space into a 1D array, as we would loose information about the proximity and similarity among the pixels' location. Additionally, most of the classical approaches were developed to capture linear interactions within the data. This type of relations do not always reflect real world scenarios, where non-linear dependencies occur between patterns and should be detected to enable a meaningful understanding [86]. Kernel based approaches have been mostly motivated by the necessity to overcome these limitations.

Kernel methods will be at the foundation of our analyses in Chapters 3 and 4. We will explore how kernels can be incorporated in classical machine learning models to learn similarity measures and solve classification and regression problems. These methods will be applied on brain MRI data and extended to combine multiple modalities, by capturing complementary characteristics from heterogeneous data sources.

The versatility of the Weisfeiler–Lehman (WL) kernel scheme 2.2.5, and the current limitations of the $\mathcal{R}$-convolution framework 2.1.3, motivated us through the development of the Wasserstein Weisfeiler-Lehman (WWL) kernel (Chapter 5). We will formulate a general Weisfeiler–Lehman propagation scheme, to generate node and graph embeddings from arbitrarily attributed graphs. Optimal assignment theory will play a crucial role to obtain similarity matrices that are sensitive to the difference in node distributions, as opposed to the $\mathcal{R}$-convolution framework which only accounts for local structural similarities.

We will conclude this chapter with an overview on graph neural networks (GNNs), considering their relation with the label refinement scheme. As for most deep learning based methods, the performance of GNNs suffers in small sample size regimes, due to the risk of overfitting. Nevertheless, it is yet unclear to which extent it is more con-

venient to use GNNs versus graph kernels. On one hand, the GNN models have the advantage of *learning* the update and aggregation function, while accounting for non-linear interactions in the data, therefore potentially have the ability to capture more subtle differences in substructures. On the other hand, especially in low sample size regimes, the GNN has the high risk to overfit, ultimately learning a model that is over representative of the training instances. On the runtime perspective, there are also conflicting visions. The GNN has a minimum required training time due to the backpropagation step, while efficient graph kernels can be very fast for small and sparse graphs. However, while GNNs are still tractable in big data regimes, computing kernels becomes infeasible for large and dense graphs. Finding the best trade-off between the usage of graph kernels and graph neural networks is yet an open research questions. We will discuss a related problem in Chapter 6, integrating GNNs with tools from domain adaptation and adversarial training, hence addressing the issue of data limitation from a transfer learning perspective.

2.1 An overview of kernel methods

Broadly speaking, a kernel is a dot product in some, possibly high dimensional, feature space. By virtue of the so-called *kernel trick*, many algorithms can be reformulated to work on the *kernel space* defined by the dot product, such that the feature representation does not have to be explicitly computed. Intuitively, this *implicit* kernel representation is a measure of similarity between pair of objects in the explicit feature space. Additionally, the similarity matrix can be directly inferred from structured data, by combining relational and value based information. We will further extend this idea in the next section, when presenting the $\mathcal{R}$ – convolution framework [82], which allows to derive similarity measures between objects by aggregating information from sub-parts of the original data. In the remainder of this section, we will present the mathematical rigorous required to understand kernel methods and introduce the notation and concepts that we will used throughout this thesis.

2.1.1 Reproducing Kernel Hilbert Spaces

Before diving into the formal characterisation of kernels, we introduce one of the crucial ingredients to their construction, the *Reproducing Kernel Hilbert Spaces*. Let us assume we have a pair of instances in some input space, i.e. $x, x' \in \mathcal{X}$. The implicit kernel representation denoted as $k(x, x') = \langle \phi(x), \phi(x') \rangle$, is defined as an inner product between the explicit feature map $\phi : \mathcal{X} \mapsto \mathcal{H}$. In order for k to be well defined[1] the feature space $\mathcal{H}$ has to be a *vector* space endowed with a dot product, more precisely it has to be an *Hilbert space*.

[1] The meaning of a well defined kernel will be clarified in the next sections; for the moment it is enough to note that a kernel has to satisfy certain properties in order to be a valid input for the machine learning algorithm.

Definition 2.1 (Hilbert space)**.** An *Hilbert space* over a vector field ($\mathbb{R}$ or $\mathbb{C}$) is an inner product space which induces a complete metric space. A metric space is complete if every Cauchy sequence is convergent.

Definition 2.2 (Cauchy sequence)**.** A sequence $x_1, x_2, \ldots$ of elements in $\mathcal{H}$, equipped with a norm $\|\cdot\|_{\mathcal{H}}$ is said to be a *Cauchy sequence* if for every $\epsilon > 0$ there exists q such that for all $i, j \geq q$, $\|x_i - x_j\| < \epsilon$.

A reproducing kernel Hilbert space (RKHS) is a particular instance of the Hilbert space defined above, with the further requirement that the set of functions evaluated at each point $x \in \mathcal{X}$ are a continuous linear functional.

Definition 2.3 (Reproducing Kernel Hilbert Space)**.** Given a non empty set $\mathcal{X}$ and an Hilbert space of functions $f \in \mathcal{H}$ where $f : \mathcal{X} \mapsto \mathbb{R}$, we say that $\mathcal{H}$ is a *reproducing kernel Hilbert space* (RKHS) is there exist a function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, such that:

(i) the kernel k has the reproducing property, i.e.

$$f(x) = \langle k(x, \cdot), f \rangle \text{ for all } f \in \mathcal{H} \tag{2.1}$$

with

$$k(x, x') = \langle k(x, \cdot), k(\cdot, x') \rangle; \tag{2.2}$$

(ii) the space $\mathcal{H}$ is spanned by k, therefore $k(x, \cdot) \in \mathcal{H}$ for every $x \in \mathcal{X}$.

2.1.2 Kernels

We now extend these concepts to characterise the kernel as a class of functions, representing similarities scores between objects.

Definition 2.4. Let $\mathcal{X}$ be a non empty set. Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a function associated with a reproducing kernel Hilbert space $\mathcal{H}$, such that there exists a map $\phi : \mathcal{X} \mapsto \mathcal{H}$ satisfying

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \tag{2.3}$$

for all $x, x' \in \mathcal{X}$. Then, we say that k is a *kernel*.

As a crucial highlight from Definition 2.4, we remark that the function ϕ does not have to be explicitly computed, indeed it is often an high dimensional representation and is infeasible to deal with it. Kernel methods rely solely on the function k to perform inference and prediction on the data $x \in \mathcal{X}$. In practice, we require k to satisfy additional properties to be well defined and valid to be used in kernel based algorithms, specifically k has to be a *positive definite (PD) kernel*. The concept of a PD kernel is strictly related to the analogous PD matrix.

Definition 2.5 (Positive definite matrix)**.** Given a real-valued symmetric matrix $K \in n \times n$, the following statements are equivalent:

(i) for all $c_i \in \mathbb{R}$;

$$\sum_{i,j} c_i c_j K_{ij} \geq 0 \tag{2.4}$$

(ii) the eigenvalues of K are nonnegative, i.e.

$$z^t K z > 0, \text{ for all non zero } z \in \mathbb{R}^n; \tag{2.5}$$

(iii) K is *positive definite*.

It is easy to deduct from Definitions 2.4 and 2.5, that applying a kernel k to every pair of instances x_i, x_j in a finite space $\mathcal{X}$, $|\mathcal{X}| = n$ gives rise to a *symmetric* matrix K encoding the corresponding kernel value, such that $K_{i,j} = k(x_i, x_j)$, for all $i, j = 1, \ldots, n$. The matrix K is also called *Gram matrix*.

Definition 2.6 (Gram matrix)**.** Let $x_1, \ldots, x_n \in \mathcal{X}$ with kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Then, the matrix

$$K = k(x_i, x_j) \tag{2.6}$$

for all $i, j = 1, \ldots, n$ is called the *Gram matrix* or *kernel matrix* of k with respect to $x_1, \ldots, x_n$.

Combining Definitions 2.4 to 2.6 we can reformulate the notion of a *positive definite kernel*.

Definition 2.7 (Positive definite kernel)**.** Let k be a kernel as per Definition 2.4. If k gives rise to a positive definite Gram matrix K, then k is a *positive definite kernel* with associated kernel matrix K.

Strictly speaking, positive definiteness as deduced from Definition 2.5 also requires $c_i = 0$ for all i. This condition is often not necessary in machine learning, and is relaxed to result in positive semi-definite kernels (PSD). For the purpose of this thesis, we will indistinguishably talk about PD and PSD kernels, with reference to Definition 2.5. From now on, we will also refer to kernels and implying that these are PSD, if not mentioned otherwise.

It can be shown that PD kernels are *closed* under certain properties, including sum, product, and multiplication for a scalar. Given two kernels K_1 and K_2, we define the following composite matrices:

(i) $K_{sum} = K_1 + K_2$

(ii) $K_{prod} = K_1 \times K_2$

(iii) $K_{sc} = cK_1$

If K_1 and K_2 are kernels, then K_{sum}, K_{prod}, and K_{sc} are also kernels for every $c \in \mathbb{R}$.

An important consequence of using k as defined in Definition 2.4, is that any machine learning algorithm relying on the dot product can be "kernelised" and formulated in

terms of the kernel. This substitution is called *kernel trick* and many of the most popular algorithms such as Support Vector Machines or k-Nearest Neighbour fall into this framework [158]. The kernel trick allows to apply these algorithms via the gram matrix only, without computing the explicit feature representation $\phi(x)$. This scheme permits the extension of numerous machine learning models on complex data domains, such as graphs and strings, that lack a natural vectorial representation. We will later discuss how the closure properties play a crucial role, as they help to generate and combine kernels to define novel similarity measures. We will now introduce some of the most popular and established kernel functions, which are well-known to produce valid kernels, while modelling different type of relations within the data. In the following we assume $x, x' \in \mathcal{X} \subset \mathbb{R}^p$ are p – dimensional instances defined in some finite subset of the real valued space.

DIRAC KERNEL. The *Dirac kernel* (or delta kernel) is possibly the most simple kernel function, aiming to assess if two objects are the same:

$$k_\delta(x, x') = \begin{cases} 1, & \text{if } x = x' \\ 0, & \text{else} \end{cases} \tag{2.7}$$

LINEAR KERNEL. The *linear kernel* is also a very simple and popular representation, defining the similarity in terms of linear interactions via a dot product:

$$k_{lin}(x, x') = \langle x, x' \rangle. \tag{2.8}$$

POLYNOMIAL KERNEL. To take into account higher order interactions between data, the *polynomial kernel* can be used:

$$k_{poly}(x, x') = (\langle x, x' \rangle + c)^d, \tag{2.9}$$

with an additive factor c and the degree of the polynomial d, as kernel parameters.

GAUSSIAN RADIAL BASIS FUNCTION (RBF) KERNEL. The *RBF kernel* (or Gaussian kernel) models a gaussian relationship between the samples and is defined as:

$$k_{RBF}(x, x') = \exp(-\gamma \|x - x'\|^2), \tag{2.10}$$

with kernel parameter γ.

SIGMOID KERNEL. Finally, the *sigmoid kernel* defines the similarity via a non-linear hyperbolic activation function:

$$k_{sig}(x, x') = tanh(\gamma \cdot \langle x, x' \rangle + c), \tag{2.11}$$

for kernel parameters c and γ.

2.1.3 $\mathcal{R}$-CONVOLUTION FRAMEWORK

There exist plenty of possibilities to construct valid kernels starting from the standard, well established, functions. Exploiting the closure properties, for instance, allows to combine different kernels providing a complementary source of information. A convenient approach, particularly suitable for complex structured data such as strings, graphs, or trees is the $\mathcal{R}$ – convolution framework proposed by Haussler [82]. The underlying idea relies on applying well known kernels on substructures of the original data, then aggregate the results to define the similarity measure on the entire entity. For this method to be valid, it is required that the object can be decomposed into a finite set of parts or substructures which, properly combined, retrieve the original instance.

To formalise the intuition behind the $\mathcal{R}$ – convolution framework, let $x \in \mathcal{X}$ be a data object having the "composite property", i.e. assume that x can be divided into D "parts" $\{x_1, \ldots, x_D\}$, where $x_d \in \mathcal{X}_d$, and $1 \leq d \leq D$. Therefore, we can represent the relation for a specific object $x \in \mathcal{X}$ and its sub-parts x_d in terms of the general relation between $\mathcal{X}_d$ and $\mathcal{X}$. Let $\bar{x} = \{x_1, \ldots, x_D\}$ and denote by $R(\bar{x}, x)$ the relation: "$\{x_1, \ldots, x_D\}$ are part of x". We further define the inverse relation as $R^{-1}(x) = \{\bar{x} : R(\bar{x}, x)\}$ and we say that R is finite if $R^{-1}(x)$ is finite, that is there are finitely many parts x_d. We observe that a relation and its inverse uniquely define a decomposition of the object x into a finite set of sub-parts.

Definition 2.8 ($\mathcal{R}$ – convolution)**.** Suppose we have two objects $x, x' \in \mathcal{X}$ such that $\bar{x}, \bar{x}'$ are the parts of x, x', respectively. Assume there exist a kernel $k : \mathcal{X}_d \times \mathcal{X}_d \mapsto \mathbb{R}$ on each $\mathcal{X}_d$ generating kernel matrices $K_d = k(x_d, x'_d)$, for each $d = 1, \ldots, D$. Then we define a the $\mathcal{R}-$ convolution kernel of $K_1, \ldots, K_D$ from the kernel function k as,

$$K(x, x') = \sum_{\bar{x} \in \mathcal{R}^{-1}(x), \bar{x}' \in \mathcal{R}^{-1}(x')} \prod_{d=1}^{D} k(x_d, x'_d). \tag{2.12}$$

From the closure properties, it follows that the kernel in equation 2.12 is a valid PD kernel, given that k is a PD kernel. The $\mathcal{R}$ – convolution kernel defined in terms of the decomposition relation introduced above can be applied to many different data domains, from strings to tuples. In the next section, we will exploit it for our application of interest on graph structured data.

2.2 GRAPH KERNELS

Graphs are ideal candidates to fit into the $\mathcal{R}$ – convolution framework, given their intrinsic modular structure. In a nutshell, these approaches rely on a graph decomposition into different substructures, such an paths, walks, or trees, followed by an aggregation step, most commonly average or sum. From Definition 2.8, it follows that a complete convolution based approach would split the graph over all its possible subparts. This results in a very high computational complexity, due to the exponential growth in the number of substructures necessary to reconstruct the graph. Graph kernels are de-

signed to find smart ways to simplify the decomposition, restricting the computation to a limited and informative number of subparts. To navigate into the abundance of existing graph kernels, we will define criteria to categorise these approaches into subclasses, depending on the type of substructure used to develop the kernel. On a successive level, it is also crucial to distinguish between the kind of graphs that these methods can handle, for instance node and edge attributes, or directed versus undirected graphs. Within the scope of this thesis, we do not aim to provide and exhaustive description of all the existing graph kernel methods, but rather present a general overview and focus on the most relevant instances.

2.2.1 Preliminaries on graphs

Before diving into the description of the different graph kernel methods, we recall the basic definitions and introduce our terminology and notation on graphs.

Definition 2.9 (Graph)**.** We define a *graph* as a tuple $G = (V, E)$ with vertices $|V| = n$ and edges $|E| = m$. If an edge exist between nodes $u, v \in V$ we denote it as $e_{u,v} = (u, v)$. A graph G is said to be *directed* if the pair (u, v) is ordered; otherwise, we say that G is *undirected*.

Definition 2.10 (Adjacency matrix)**.** Given an undirected graph $G = (V, E)$ the *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ uniquely determines the topology of the graph. It is defined as $A_{i,j} = a_{i,j}$, with $i, j = 1, \dots n$, such that $a_{i,j} = 1$, if e_{v_i,v_j} exists, $a_{i,j} = 0$, else.

The concept of *neighbourhood* is one of the most fundamental in graph theory. Generally speaking, the neighbourhood of a node consists of all the nodes that are connected to it by an edge. More precisely, we also call this the 1 – neighbourhood. Then, a k – neighbourhood is defined according to the node distance within the graph.

Definition 2.11 (Node distance)**.** Given an undirected graph $G = (V, E)$ we define the *distance* between two nodes $u, v \in V$, $d(u, v)$, as the minimum number of edges necessary to reach v from u, and viceversa.

Definition 2.12 (K-neighbourhood)**.** Given a graph $G = (V, E)$ we define the k-neighbourhood of a node $v \in V$ the set of nodes that can be reached from v with at most k hops, i.e. that have distance k from v. Equivalently, a node $u \in V$ is in the neighbourhood of v if u and v are separated by at most k edges:

$$\mathcal{N}^k(v) = \{u \in V : d(u, v) \leq k\}. \tag{2.13}$$

In the following, we will refer to the *neighbourhood* of a node as its 1 – neighbourhood, unless specified otherwise

Definition 2.13 (Degree)**.** The *degree* of a node $v \in G$ in a graph, is defined as the cardinality of its 1 – neighbourhood, or equivalently as the number of outgoing edges from v, that is:

$$deg(v) = |\mathcal{N}(v)|. \tag{2.14}$$

We now have all the ingredients to define fundamental elements in graph theory, i.e. walks and paths, which are substructures of the original graph and will be especially useful to design convolution kernels.

Definition 2.14 (Walk, path, shortest path)**.** We define a *walk* w in a graph $G = (V, E)$ as a sequence of nodes $w = \{v_1, \dots, v_r\}$, $v_i \in V$ for $i = 1, \dots, r$, where consecutive nodes are connected by edges, i.e. $(v_i, v_{i+1}) = e_{v_i, v_{i+1}} \in E$ for $1 \leq 1 < r$. The length of the walk is equal to the number of edges $r - 1$. If it holds that $v_i \neq v_j \iff i \neq j$, or equivalently there are no self-loops, then the walk is called a *path*. The *shortest path* between two nodes v_i and v_j in V is the path of minimal length connecting them.

Enumerating shortest paths in a graph is a non trivial task, due to the high computational complexity from having to evaluate all the paths and their length between two nodes. In practice, shortcuts can be employed to reduce the search space. Two popular algorithms to find all shortest paths in polynomial time are Dijkstra [43] and Floyd-Warshall [53, 190].

Nodes and edges often contain additional information depending on the nature of the data. For example, the edges might be weighted according to the spatial distance between the objects (nodes). The nodes are often representative of specific entities and contain an associated feature representation. This information is encoded within the graph structure in the format of attributes or labels.

Definition 2.15. Let $G = (V, E)$ be a graph such that $|V| = n$ and $|E| = m$. We characterise node and edge attributes and labels as follows.

(i) The graph G is *node attributed*, or simply *attributed*, if there exist an embedding function $\ell : |V| \mapsto \mathbb{R}^p$ such that $\ell(G) = X_G \in \mathbb{R}^{n \times p}$. We call X_G the *node feature* matrix, where each row $i = 1, \dots, n$ contains the *node attributes* of node v_i. In the special case where $p = 1$ and $\mathbb{R} = \mathbb{N}$ the attributes are categorical, we refer to them as *labels* and to G as a categorically labelled, or simply *labelled* graph.

(ii) We say that G is *edge attributed* if there exist a function $w : |E| \mapsto \mathbb{R}^q$ such that $w(G) = W_G \in \mathbb{R}^{m \times q}$. We call W_G the edge feature matrix, where each row $i = 1, \dots, m$ contains the *edge attributes* of edge e_i. In the special case where $q = 1$ and the attributes express a measure of similarity between their end nodes, we refer to them as *weights*, to W_G as the weight matrix, and we say that G is a *weighted* graph.

Throughout the text, when referring to an *attributed* graph we implicitly consider it to be node attributed, unless specified otherwise. We will also use the terms *labelled* and *categorically labelled* indistinguishably.

GRAPH ISOMORPHISM PROBLEM. Intuitively, one can define a criteria for equivalence between graphs based on their topological structure. This is called the *graph isomorphism* problem, which is NP and, as of today, no algorithm is known to solve it in polynomial time [62].

Definition 2.16 (Graph isomorphism)**.** Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. They are said to be *isomorphic* if there exist a bijection between their nodes. More formally, a *graph isomorphism* between G and G' is a bijective function $f : V \mapsto V'$ preserving adjacency, i.e. if $(u, v) \in E$ are adjacent in G, then $(f(u), f(v)) \in E'$ are adjacent in G'.

The criteria for isomorphism defines a similarity measure between graphs. However, this would be computationally infeasible, poorly scaling with the number of nodes. Furthermore, similarity measures based on exact isomorphism could be too restrictive, since two graphs will be considered similar if and only if their structures exhibit exact matching [161]. Overcoming these issues, was one of the main motivation for the exploration of graph kernels. Gärtner *et al.* [66] observed that computing a similarity over all substructures of the graph, is equivalent to check if they are isomorphic. Then, graph kernels were developed as a family of graph comparison algorithms, which evaluated partial similarities by limiting the search space to a finite set of substructures.

2.2.2 Graph kernels based on nodes or edges

The simplest entities composing graphs are nodes and edges, as well as their attributes. Then, the most naive kernels we can construct are based on these objects.

The node kernel

Graph kernels based on nodes define a measure of similarity that ignores the edge structure, therefore neglecting the graph topological information. These kernels, while being very simplistic, can be of help for certain scenario, in particular: (1) they represent an excellent baseline to evaluate the effectiveness of different methods that exploit more complex graph structures; (2) they can show very good performance if the necessary information is encoded in the attributes. We consider two kind of node based graph kernels. A more general one that is suitable for all type of attributed graphs, the *all node-pairs kernel*, and another formulation that is particularly designed for labelled graphs, the *node histogram kernel*. The all-node pairs kernel is an $\mathcal{R}$ – convolution inspired framework that compares node attributes pairwise and subsequently aggregate those to derive a kernel similarity.

Definition 2.17 (All node-pairs kernel)**.** Given two attributed graphs $G = (V, E)$ and $G' = (V', E')$, we define the *all node-pairs kernel* as:

$$K_N(G, G') = \sum_{v \in V} \sum_{v' \in V'} k_{node}(v, v'). \tag{2.15}$$

In the above formulation, $k_{node}(v, v')$ is a valid base kernel, for example linear or RBF, defined between node attributes:

$$k_{node}(v, v') = k_{base}(x_v, x_{v'}) \tag{2.16}$$

where x_v and $x_{v'}$ are the corresponding rows in the node feature matrices X_G and $X_{G'}$, respectively.

For special cases of x_v and k_{node}, it is easy to find the corresponding explicit graph feature representation and compute the kernel via inner product, saving considerably runtime. For example, if the graphs have categorical node labels and $k_{base} = k_{lin}$, then

$$K_{N(G,G')} = \langle \phi(G), \phi(G') \rangle, \tag{2.17}$$

where $\phi(G) = \sum_{v \in V} x_v$ and similarly $\phi(G') = \sum_{v' \in V'} x_{v'}$.

For categorical node labels, we can also define a kernel by creating histograms of the node labels.

Definition 2.18 (Node histogram kernel)**.** Let G, G' being categorically labelled graphs, such that $\ell : \{V, V'\} \mapsto \Sigma_{V,V'}$, i.e. $\Sigma_{V,V'}$ is the joint alphabet of node labels of G and G'. We denote by $\phi(G)$ and $\phi(G')$ the histogram of node labels in G, G'. Then,

$$K_{NH}(G, G') = k_{base}(\phi(G), \phi(G')). \tag{2.18}$$

Particular choices of the kernel and node labels lead to specific formulation of the node kernels. In particular, one can verify that:

(i) If k_{base} is the linear kernel, then $\phi(G)$ and $\phi(G')$ are the explicit feature representation of G and G' for kernel K_{NH}.

(ii) If k_{node} is a Dirac kernel and k_{base} is *linear* then $k_N = k_{NH}$.

(iii) Complexity of K_N is $\mathcal{O}(n^2 p)$ and of K_{NH} is $\mathcal{O}(np)$, where p is the attribute dimension and node is the number of samples.

THE EDGE KERNEL

As for the node, the edge kernels are very useful baselines when developing new methods and can be exploited to assess the relative impact of the edge information. Edge kernels can be defined analogously to node kernels.

Definition 2.19 (All edge-pairs kernel)**.** Given two attributed graphs $G = (V, E)$ and $G' = (V', E')$, we define the *all edge-pairs kernel* as:

$$K_E(G, G') = \sum_{e \in E} \sum_{e' \in E'} k_{edge}(e, e'), \tag{2.19}$$

where $k_{edge}(e, e')$ is any base kernel (e.g. linear or RBF) defined between edge attributes.

Definition 2.20 (Edge histogram kernel)**.** Let G, G' being such that the edge attributes are categorical and denote by $\phi_E(G)$ and $\phi_E(G')$ the histograms counting the occurrence of edge labels in G, G'. Then,

$$K_{EH}(G, G') = k_{base}(\phi_E(G), \phi_E(G')). \tag{2.20}$$

2.2.3 Graph kernels based on walks and paths

Despite the attractiveness of node and edge kernels, due to their simplicity and straightforward interpretation, they lack the ability to capture the complete topology of the graph and the relation among distant sub-entities. Enabling the use of sub-structures as walks and paths leads to a wider range of possibilities for kernels generation. We begin by considering walks as sub-parts of the original graph: comparing two graphs by evaluating every possible walk between node pair is known to be NP-hard, as it is computationally equivalent to solving the graph isomorphism problem [66]. To overcome these difficulties, kernels defined on fixed length random walks and on label matching strategies between walks have been proposed [66, 97]. The *random walk kernel* is based on the idea of counting occurrences of label sequences of a certain length in the *direct product graph*, which is constructed by connecting pairs of vertices in G and G', if they are both connected in the original graph [66]. While this can only be applied on categorically attributed graphs, at the same time, Kashima *et al.* [97] proposed a generalisation on graphs with continuous node and edge attributes. Both methods have been shown to be computable in polynomial time, however the runtime depends on the number of nodes n as $\mathcal{O}(n^6)$, making the computation practically infeasible on large size graphs.

Following the sharp development of random walk kernels, two phenomenons have been discovered to negatively affect their empirical performance and theoretical effectiveness, *tottering* and *halting*. By definition (see Definition 2.14), walks are allowed to visit the same node multiple times, then they can be "stuck" in a cycle leading to very high similarities values for graphs that might have only a few nodes and edges in common. Such phenomenon is known as *tottering*. An extension of the *marginalised graph kernel* has been proposed to prevent this issue by adjusting the probability of revisiting the same node in the random walk process [118]. *Halting* can be empirically observed in *geometric random walk kernels* [20], referring to the problem that long walks are downweighted due to the exponential decay employed in the computation of the similarity score. This results in the kernel values being dominated by walks of unitary length. Sugiyama and Borgwardt [167] studied this phenomenon and proposed a *k-step random walk kernel*, which alleviates the problem by fixing the weight parameter and upper bounding the length of the walk with a limited number of steps.

Shortest path kernel

A different line of research was devoted to address the drawbacks of walk kernels, embracing the potential of replacing *walks* by *paths*. It is worth to note that by using *paths* instead of *walks* the problem of tottering disappears, since a path cannot visit the same node or edge twice (see Definition 2.14). The *shortest path kernel* [21] is in spirit similar to a random walk kernel, evaluating the similarity between graphs by aggregating a score matching of their shortest paths. More formally, given a graph $G = (V, E)$ the first step is to compute the transformed shortest path graph as $S = (V, E^S)$, where S has the same nodes as G and the edges are labelled by the length of the shortest path between their end nodes in G.

Definition 2.21 (Shortest-path kernel)**.** Let $G = (V, E)$ and $G' = (V', E')$ being attributed graphs and let $S = (V, E_S)$ and $S' = (V', E'_S)$ be their shortest-path transformation, then we define the *shortest path kernel* as

$$K_{SP}(G, G') = \sum_{e \in E_S} \sum_{e' \in E'_S} k_{walk}(e, e') \tag{2.21}$$

where k_{walk} is a kernel on edge walks of length 1 defined on the transformed shortest path graphs.

For example, in the case of categorically labelled graphs, one can define:

$$k_{walk(e,e')} := k_{node}(u, u') \cdot k_{edge}(e, e') \cdot k_{node}(v, v'), \tag{2.22}$$

where $e = (u, v)$ and $e' = (u', v')$. In particular, by setting k_{node} as the Dirac kernel, we obtain

$$k_{walk}(e, e') = \begin{cases} 1, & \text{if } \ell(u) = \ell(u') \wedge \ell(v) = \ell(v') \\ 0, & \text{else} \end{cases} \tag{2.23}$$

Within this setting, K_{SP} values are obtained by counting occurrences of paths with equal start and end point node labels. In general, the formulation of the shortest path kernels allows for applicability on various type of graphs, with both node and edge attributes. However, the implicit formulation in Definition 2.21 has a runtime of $\mathcal{O}(n^4)$, since the kernel must be evaluated between every pair of nodes. Nevertheless, for the categorically labelled case, an explicit feature representation has been derived resulting in a drastic runtime improvement to $\mathcal{O}(n^2)$ [104].

A speed-up extension of the shortest path kernel on continuously attributed graphs, relies on limiting the search space to paths with the same length. This approach is known as *GraphHopper kernel* and enjoys an equivalent runtime as the explicit shortest path of $\mathcal{O}(n^2 p)$, where p is the dimension of the attributes [50].

2.2.4 Graph kernels based on sub-graphs

Despite their expressiveness, walks and paths only incorporate a selected type of the graph information. A more expressive approach would be to use subgraphs of arbitrary type which ultimately can fully represent the graph. However, enumerating all possible sub-graphs degenerates to the graph isomorphism problem, while simultaneously having the issue of getting a similarity measure that overfits on the selected graph. The *Graphlet* kernel [160] aims at exploiting sub-graphs of arbitrary shape, but of a limited size, alleviating both the overfitting and complexity issue. Similarly as for the walks and paths, a kernel can be defined by counting the occurrences of selected *graphlets* in the graphs.

2.2.5 Graph kernels based on iterative label refinement

A novel and successful class of graph kernels was introduced by Shervashidze and Borgwardt [159] based on the idea of iterative label refinement. In a nutshell, these methods construct a multi-iterative set of graphs, where the representation at each iteration is an updated version of the previous one, starting with the original graph at iteration zero. As the number of iterations evolve, more topological information is incorporated into the graph and after h-steps the information on the h-hop neighbourhood is included.

Definition 2.22 (Label refinement)**.** Let us consider a graph $G = (V, E)$, for simplicity we assume that G is categorically labelled and without edge attributes. Let $\ell(v_i^0)$ be the label of node v_i, for each $v_i \in V$, with $i = 1, \ldots, |V|$. Then, we define the *label refinement* of graph G at step h via a node label update, as:

$$\ell(v_i^h) = f(\ell(v_i^{h-1}), g(\ell(u_i^{h-1}) : u_i \in \mathcal{N}(v_i))), \tag{2.24}$$

for arbitrarily chosen functions f and g.

The specific formulation of f and g generates different kind of graph kernels. We will see that this label refinement step is extremely close to the modern Graph Neural Networks (GNNs) approaches, where f is typically a non-linear activation function and g incorporates a weighted combination with learned parameters. The relationship between graph kernels and GNNs will be explored in Section 2.3. In the following, we will discuss the most popular label refinement kernel methods.

The Weisfeiler-Lehman kernel framework

Inspired by the Weisfeiler–Lehman test of isomorphism [192], the corresponding kernel based framework achieved outstanding empirical performance in graph classification and has built the foundation for many existing methods. Intuitively, the Weisfeiler-Lehman test of isomorphism relies on the concept of iteratively creating multisets, i.e. a sorted string consisting of the label of each node and its neighbours, to be hashed to new node label. This procedure is repeated until the desired number of iterations or until no label update is performed. Ultimately, the comparison among sequence of compressed labels establishes a criteria for graph isomorphism, which we now state without proof; additional details can be found in [10].

Proposition 2.1. *Given two graphs G and G' and their corresponding sequence of compressed labels, if the sequences are different we conclude that G and G' are non-isomorphic; if the sequences are equal, then the graphs are likely to be isomorphic [10].*

Proposition 2.1 implies that the equality of label sequences is necessary, although not sufficient, to conclude that two graphs are isomorphic.

The Weisfeiler–Lehman (WL) kernel follows the same iteration scheme. Graph features are obtained by aggregating the compressed labels sequence and input to a linear kernel to get a similarity value. More precisely, the WL kernel creates a label sequence

for each node and iteration, where at each iteration the new label is defined by a unique hash of the ordered string composed by current node labels and those of its neighbours. Mathematically, this corresponds to replacing f by a perfect *hash* function and g with the identity in Equation 2.24, i.e.:

$$\ell^{h+1}(v) = \mathsf{hash}(\ell^h(v), \mathcal{N}(v)). \tag{2.25}$$

We refer to the iterative procedure generated by equation 2.25 as the WL labelling refinement scheme. As a consequence of the perfect hashing, two nodes at iteration $h+1$ will have the same label if and only if both their label and those of their neighbours at iteration h are equal. In the general formulation, the WL kernel framework defines a sum of kernels, where each contribution is given by the kernel between graphs at the different iterations. The most used instance of the WL framework is the WL *subtree kernel*, which employs histograms of node labels at multiple iterations to derive a graph feature representation.

Let $\mathcal{G} = \{G^0 = (V, E, \ell^0(V)), \dots, G^H = (V, E, \ell^H(V))\}$ be the sequence of graphs obtained by the WL labelling scheme. With a slight abuse of notation, let us denote by $|\ell^h(V)| = L$ the cardinality of the node labels at iteration h, that is the number of unique node labels at each iteration. We define the histogram of graph G at iteration h as $\phi^h(G) = [|\ell_0^h|, \dots, |\ell_L^h|]$, where $|\ell_j^h|$ is the number of occurrences of label ℓ_j in graph G^h. Then, we call $\phi(G) = (\phi^0(G), \dots, \phi^H(G))$ as the concatenation of features at different iterations, the WL feature representation.

Definition 2.23 (WL subtree kernel)**.** Given two graphs G, G' with WL graph feature representations $\phi(G)$ and $\phi(G')$, the WL subtree kernel is defined as

$$K_{WL-subtree}(G, G') = \langle \phi(G), \phi(G') \rangle \tag{2.26}$$

We will often denote $K_{WL-subtree} = K_{WL}$. We can express the WL subtree kernel using explicit node and graph feature representation, that will provide us with a direct connection with the GNN discussed in Seciton 2.3

Definition 2.24 (Node WL feature)**.** For a graph G, let $\ell^0(V)$ be the original node labels associated to it. Let $\mathcal{A}_\ell^0$ be the alphabet of original node labels; by analogy, we define $\ell^h(V)$ and $\mathcal{A}_\ell^h$ the corresponding labels and alphabet at WL iteration h. We define a node feature associated to a node $v \in V$ at some iteration h as:

$$x_{WL}^h(v) = \mathsf{onehot}(\ell^h(v)), \tag{2.27}$$

that is $x_{WL}^h(v) = [x_{0,WL}^h(v), x_{1,WL}^h(v), \dots, x_{|\mathcal{A}_\ell^h|-1,WL}^h(v)] \in \mathbb{N}^{|\mathcal{A}_\ell^h|}$ satisfies

$$x_{i,WL}^h(v) = \begin{cases} 1, & \text{if } \ell^h(v) = i \\ 0, & \text{else} \end{cases} \tag{2.28}$$

By definition of histogram and of $\phi(G)$, it follows that

$$\phi^h(G) = \sum_{v \in V} x^h_{WL}(v) \tag{2.29}$$

and

$$\phi(G) = [\phi^0(G), \ldots, \phi^H(G)]. \tag{2.30}$$

Therefore, it is possible to explicitly compute the WL subtree kernel (in Equation 2.26) by a sum of node WL features which generate the graph feature itself. In terms of the $\mathcal{R}$ – convolution framework, the WL subtree kernel can be seen in a sum aggregating fashion, where the *parts* of the graph are represented by the node features obtained via the WL scheme at multiple iterations. In practice, a node and its neighbourhood are seen as substructures. We will discuss in Chapter 5 how these perspective of WL allows us to extend the existing framework to continuously attributed graphs. We will also replace the sum aggregation with a more complex function, to better capture the similarities between distribution of node labels.

Plenty of new kernels have been developed to extend the original WL framework in the subsequent years. The *neighbourhood hash kernel* [84] was proposed in parallel to the WL kernel and is based on a similar label refinement scheme. The core difference between the two approaches appears in the hashing step, which is *non-perfect* in the latter one, implying that collisions can occur. This is achieved by representing updated labels via binary strings of fixed length, which could then be mapped to the same value despite their ordered string being different. This approach benefits from reduced runtime, at the cost of expressivity. An improvement that aimed at optimising the trade-off between expressivity and efficiency came several years later with the Hadamard code kernel, which reduces the expected amount of collisions by introducing a special encoding scheme [98]. A common limitation of these iterative label refinement kernels is the lack of applicability to continuously attributed graphs; the *propagation kernel* [130] and *hash graph kernels* [126] explore this direction.

2.2.6 Beyond the $\mathcal{R}$–convolution framework

At the beginning of this section, we mentioned that most graph kernels rely on the $\mathcal{R}$ – convolution framework, performing a simple aggregation step, such as sum or average, across substructure similarities. However, one might be interested to find partial overlaps, i.e. optimal matches (assignments) between subparts of the graph [57]. The first optimal assignment kernel by Fröhlich *et al.* [57] was later shown not to be guaranteed as positive definite, leading to theoretical complications for the classificaiton learning algorithm [182]. Recently, Kriege *et al.* [105] proposed an optimal assignment kernel based on label refinement features obtained with WL and proved it to be positive semidefinite.

Definition 2.25 (Optimal assignment kernel)**.** Given two graphs $G = (V, E)$, $G' = (V', E')$ categorically labelled, such that $V, V' \in \mathcal{A}$, the alphabet of all potential node

labels, let $f(V, V')$ be the set of all bijections between their nodes. Suppose k_{node} is a base kernels on the set of vertices, then the *optimal assignment kernel* is defined as

$$K_{OA}(G, G') = \max_{f \in f(V,V')} \sum_{v \in V, v' \in V'} k_{node}(v, v'), \tag{2.31}$$

with $k(G, G') = 0$ if $|V| \neq |V'|$.

Kriege *et al.* [105] showed that if k_{node} arises from a hierarchical partition of the kernel domain, then K_{OA} is positive semidefinite. Choosing k_{node} as a Dirac kernel and defining the hierarchical structure via the Weifeiler–Lehman iterations leads to a new family of graph kernels, the *Weifeiler–Lehman optimal assignment kernel* (WL-OA). Following the notation for the Weisfeiler–Lehman scheme (see Section 2.2.5), WL-OA between $G = (V, E)$ and $G' = (V', E')$ is defined between their nodes while using a base kernel that evaluates the compressed labels, i.e. the WL node features $x^h_{WL}(v)$ at multiple iterations:

$$k_{node}(v, v') = \sum_{h=0}^{H} k_\delta(x^h_{WL}(v), x^h_{WL}(v')), \tag{2.32}$$

where k_δ is a Dirac delta kernel and $v, v' \in V, V'$.

2.3 Graph Neural Networks

The late explosion of deep learning has influenced nearly every subfield of machine learning and data mining, and graph representation learning is not an exception. The earliest Graph Neural Network (GNN) was proposed more than a decade ago by Gori *et al.* [74] and later extended in Scarselli *et al.* [155]. These approaches rely on a node feature update via Recurring Neural Networks (RNN) and can be considered as precursors of the recent deep learning based GNNs. The main difference with modern methods is in the training procedure, which in the original work was performed until convergence, rather than via fixed number of iterations or early stopping criteria [30]. Subsequent attempts to generalize Convolutional Neural Networks (CNN) employed spectral graph theory to model the signal [26, 41]. However, these came with several limitations such as high complexity, difficulties in incorporating node and edge features, and extension to inhomogeneous structured set of graphs. Most of the modern GNNs fall under the general *message passing framework*, consisting in two main steps: (1) node or edge features aggregation; (2) feature update. The main idea is closely related to the propagation scheme presented for the WL kernels. Given an initial node or edge feature representation, the feature is updated by a weighted aggregation over connected structures (e.g. the 1 – neighbourhood) and then transformed with a non-linearity function. The most popular GNN methods include: the Neural Fingerprints Network [45], pioneer work in the field designed with the aim to *neuralize* classical circular fingerprints as molecular descriptors [70]; Graph Convolutional Networks [100], which combined spectral representation with an expressive propagation scheme; the Graph Invariant Network [198] that notably depicts the link to the graph isomorphism problem. We refer to a recent

survey for a general overview [196] and to the work of Gilmer *et al.* [69] for additional details on the message passing framework. In the reminder of this section, we will provide a formal introduction to GNNs and particularly discuss their relationship with the WL propagation scheme [69, 198].

2.3.1 The Graph Neural Network model

Consider a graph $G = (V, E)$, with $|V| = n$ and $|E| = m$; let $X_G \in \mathbb{R}^{n \times p}$ be the node feature matrix, that is $x_v^0 \in \mathbb{R}^p$ is the initial node attribute associated with node v. For each node $v \in V$ we denote by $u \in \mathcal{N}(v)$ the nodes in their neighbourhood. Given the number of layers H in the GNN, we define the recursive updating scheme at each layer $h = 1, \ldots, H$ as:

$$z_v^{h-1} = g(x_u^{h-1}) \tag{2.33}$$

$$x_v^h = f(x_v^{h-1}, z_v^{h-1}), \tag{2.34}$$

for arbitrarily chosen g and f. The function g is also called the *aggregation* function, determining the *aggregation* step of the GNN, and is usually defined as a weighted linear combination, e.g. a *sum*. The function f is typically non-linear activation, for example a *ReLU* or *tanh*, giving rise to the *update* step. We emphasise the explicit analogy between equations 2.33, 2.34 and equation 2.25: choosing g as the identity function and f as a *perfect hashing* we recover the WL refinement scheme. In other words, if the initial node feature x_v^0 corresponds to the *onehot* encoding of the node label, then it is also equivalent to the WL node feature introduced in Definition 2.24.

The GNN model is suitable for various graph based prediction tasks, including node classification, link prediction, and graph level classification and regression. In node level prediction task, the node representation at multiple layers x_v^h is sufficient to solve the classification or regression problem. Then, either a *softmax* or a *linear* layer are applied to x_v^h to perform the prediction and subsequently employed in a loss function to optimise the parameters of the GNN. In a setting of graph-level classification or regression, an additional step is required to obtain a graph embedding. Specifically, the node representations are *combined* to generate a graph-level feature as input for the *softmax* or *linear* layer. This *combination* step is defined as follows:

$$\phi^h(G) = r(\{x_v^h \mid v \in V\}) \tag{2.35}$$

$$\phi(G) = c(\{\phi^h(G) \mid h = 0, \ldots, H\}). \tag{2.36}$$

The function r in equation 2.35 is often called the *readout* function. For the map c it is common to choose a function that only keeps the latest representation $\phi^H(G)$, but in principle intermediate layers can be incorporated [45]. Again, we should point out the similarity with the WL scheme and in particular with Equations 2.29 and 2.30. Here, it is straightforward to observe that replacing r by a *sum* and c by a *concatenation*, we have an exact correspondence with the WL subtree kernel.

2.3.2 The graph isomorphism problem in GNNs

We earlier discussed how pioneer approaches in graph kernels (see Section 2.2.3, [66, 97]) explicitly addressed the issues related to the complexity of the graph isomorphism problem, and aimed at building efficient methods while maximising the expression power. Besides, the direct derivation of the WL kernel from the WL test of isomorphism, again affirms the crucial importance of establishing a connection between the graph isomorphism problem and representation learning algorithms. From Proposition 2.1 it follows that, despite equal WL sequences are *luckily* to be generated from isomorphic graphs, there exists graphs which are isomorphic and cannot be distinguished by the WL test, thus neither by the WL kernel. Therefore, it comes naturally to wonder whether modern and expressive GNNs are more powerful than the WL test in distinguishing isomorphic structures [198]. The answer is negative; indeed, the GNN framework as described in Section 2.3.1, can be *at most* as powerful as the WL test in discriminating graph structures. In other words, if two graphs G and G' are mapped into different embeddings $\phi(G) \neq \phi(G')$ by a GNN, then the WL sequences are also different. Furthermore, for the GNN to be as powerful as the WL test, the functions g and r need to satisfy specific properties.

Theorem 2.2 (Xu *et al.* [198]). *Let $\phi(G)$ and $\phi(G')$ be the graph feature representation of G and G' obtained by a GNN model, through the updating scheme (Equations 2.33, 2.34, 2.35, 2.36). If the WL test of isomorphism decides that G and G' are different then, with a sufficient number of layers, the GNN generates different embeddings, i.e. $\phi(G) \neq \phi(G')$ if the functions f, g, and r (Equations 2.33, 2.34, 2.35) are injective.*

Not every GNN satisfies the injective property, for example the very popular Graph Convolutional Network [100] does not. However, choosing f as a multi-layer perceptron (MLP; [54]) and by virtue of the universal approximation theorem [87], it can be shown that the resulting neural network is an instance of Theorem 2.2; such a model is called Graph Isomorphism Network (GIN). The node updating rule for the GIN architecture can be written as:

$$x_v^h = MLP^h\left((1+\epsilon^h)x_v^{h-1} + \sum_{u \in \mathcal{N}(v)} x_u^{h-1}\right), \tag{2.37}$$

corresponding to an aggregation function g (Equation 2.33) being a *sum* and the *update* function f (Equation 2.34) is an MLP with additional parametrization given by ϵ. The graph graph feature is obtained with a *readout* (Equation 2.35) equal to *sum* and using *concatenation* to combine different iterations (Equation 2.36):

$$\phi(G) = concatenate(\{\sum\{x_v^h \mid v \in V\}, h = 0, \ldots, H\}) \tag{2.38}$$

We refer the reader to Xu *et al.* [198] for additional technical details and the proof of Theorem 2.2.

Part II

Multi-modal multi-task analysis of neurological and psychiatric disorders

3 Classification of patients and healthy individuals using brain MRI

Characterising the morphology of the human brain and understand the elemental changes occurring within the presence of a neurological disorder, has been a major topic of investigation in the last decades. Precision medicine aims to develop efficient personalized therapies based on the clinical, historical, and genetics characteristics of the individual. Then, extracting individual significant patterns and features from clinical exams is crucial in this context. Brain imaging data is playing a central role in this development, with the acquisition of patient scans becoming more common in clinical routines and scanner machines improving their quality. However, collection of brain magnetic resonance imaging (MRI) data is still an expensive procedure, with respect to time and economical cost. Besides, it is generally an uncomfortable practice for the subject themselves. These obstacles lead to MRI studies often suffering from small sample size. Furthermore, intrinsic differences among scanner machines and inhomogeneity of procedures and acquisition strategies causes sample cohorts to be incomparable across studies. To partially overcome these limitations, the importance to establish consistent and organised databases from medical, and in particular MRI data, is arguably one of the major challenges of the XXI century. The enormous potential of new analytic algorithms, to detect hidden patterns in big data, could bring to the discovery of early signs of disease and personalised treatments. The need to collect large and organised medical data has been supported by numerous initiatives in the MRI field, creating public cohorts available for researchers and clinicians. Among the most famous, we recall the Human Connectome Project [1] (HCP; [47]), whose objective is to create a complete map of the human brain, including functional and structural connections, within and across individuals. We also recall another popular database created for the study of Alzheimer's disease: the Alzheimer's Disease Neuroimaging Initiative [2] (ADNI; [127]). The project started as a research collaboration, with the goal to collect longitudinal data, including clinical, imaging, genetic and biochemical sources to track the development of Alzheimer disease. Another powerful resource is the UK Biobank [3] [176], a national effort to improve treatment, diagnosis and prevention of multiples diseases.

[1] http://www.humanconnectomeproject.org

[2] http://adni.loni.usc.edu

[3] https://www.ukbiobank.ac.uk

The created database includes a variety of sources, for example genetic samples, clinical variable, and ultimately imaging data.

Our work can be seen in the context of a similar initiative supported by Horizon 2020[4], within a project with the ambition to develop a clinical decision support system for the analysis of multi-modal quantitative MRI data (CDS-QuaMRI[5]). A European consortium including universities, hospitals, and companies was established to bring together data and experts from several fields. The major aim was to create an homogeneous database and a software framework to apply latest machine learning techniques on novel MRI data metrics. The novel multi-modal techniques aim to overcome the boundaries of conventional MRI readouts, which have shown limited prognostic value and only partially explain disease progression and treatment response. We will investigate the predictive power and learning capabilities of several state-of-the-art machine learning algorithms on two clinical studies, from Major Depression Disorder (MDD) and Multiple Sclerosis (MS) subjects as well as matched Healthy Controls (HC). We aim to understand the effect of using a single or a combination of MRI images, and unfold their relevance with respect to several neurological tasks.

The remainder of this chapter is organised as follows. We first describe the data collection and preprocessing steps in Section 3.1, from our two separate cohorts of depressed and multiple sclerosis patients. Subsequently, we introduce the main methodologies and pipelines used for the machine learning analysis, including the prediction models (Section 3.2) and feature extraction techniques (Section 3.3). We finally report our experimental setup and findings in Section 3.4 and conclude with a discussion. We particularly focus on the limitations of our study and the relevance of multi-modal images (Section 3.5).

3.1 Data description

3.1.1 Major depressive disorder study

Major depressive disorder (MDD), also simply referred as (clinical) depression, is a mental health disorder characterised by generalised low mood and multiple associated symptoms, which may include sadness, loss of interest, anger, anxiety, sleep disturbance, weight alteration, and reduced attention. Diagnosis of MDD is performed by a clinical expert based on the presence for a prolonged period of times (usually two weeks) of one or several of these symptoms. The assessment is generally accompanied by a questionnaire and sometimes a blood test. Nevertheless, even for an expert psychiatrist, clearly define the depression phenotype is far from being trivial, due to the lack of universally established criteria. As a result, the clinical label is often uncertain, especially for the borderline patients. Furthermore, several scales to define the severity of the disorder exist, which makes the evaluation even more challenging. This confu-

[4] https://ec.europa.eu/programmes/horizon2020/en

[5] https://cds-quamri.eu/

Table 3.1: Descriptive statistics of the MDD study cohort.

	MDD ($n = 57$)	HC ($n = 61$)	Group statistics
Age (mean, std)	40.5 ± 12.7	38.3 ± 10.1	$t(116) = 1.04$, $p > 0.1$ [6]
Sex (m/f)	25/32	35/26	$c^2(1, n = 118) = 2.15$, $p > 0.1$ [7]
BDI (mean, std)	27.42 ± 8.28	NA	NA
Number of episodes (mean, std)	7.51 ± 5.43	NA	NA
Medication status [8] (med. free/on med.)	35/22	NA	NA
WM accuracy (in %; mean, std)	73.6 ± 16.5	70.3 ± 30.0	$t(115) = 0.73$, $p > 0.1$ [6]
WM reaction time (in ms; mean, std)	646 ± 1.37	590 ± 179	$t(115) = 1.90$, $p > 0.05$ [6]

sion will inevitably be reflected in the machine learning analysis, as we will discuss in Section 3.5.

Data selection

The MDD study consists of 57 patients with depression and a set of 61 matched healthy controls (HC), recruited at the Free University of Berlin (FUB) and at the University of Zuerich (UZH). The depressed patients (DP) have had at least one acute depressive episode and the MDD severity was assessed with the Back Depression Inventory (BDI) criteria [12]. The HC were screened for psychiatric disorders using the short version of the Structured Clinical Interview for Diagnostic and Statistical Manual of Mental Disorders. Subjects from both groups were excluded based on the following criteria: major medical illness, history of seizures, head trauma with loss of consciousness, and pregnancy. For MDD patients, screening was also done with respect to atypical forms of depression, suicidal ideation, any other psychiatric disorder, history of substance abuse or dependence, and Electroconvulsive Therapy (ECT) in the previous 6 months. For HC, subjects with present or previous diagnosis of any psychiatric or neurological disorder were excluded. All the participants signed a written consent before entering the study, which was carried out in accordance with the latest version of the Declaration of Helsinki [8]. Data descriptive statistics of the cohort are reported in Table 3.1.

Data acquisition and preprocessing

The acquisition and processing protocol established for the MDD sample data included MRI images from several modalities collected at the FUB and UZH. An overview of the different MRI modalities and their characteristics, as well as additional details on the preprocessing and acquisition pipelines is provided in Appendix A.

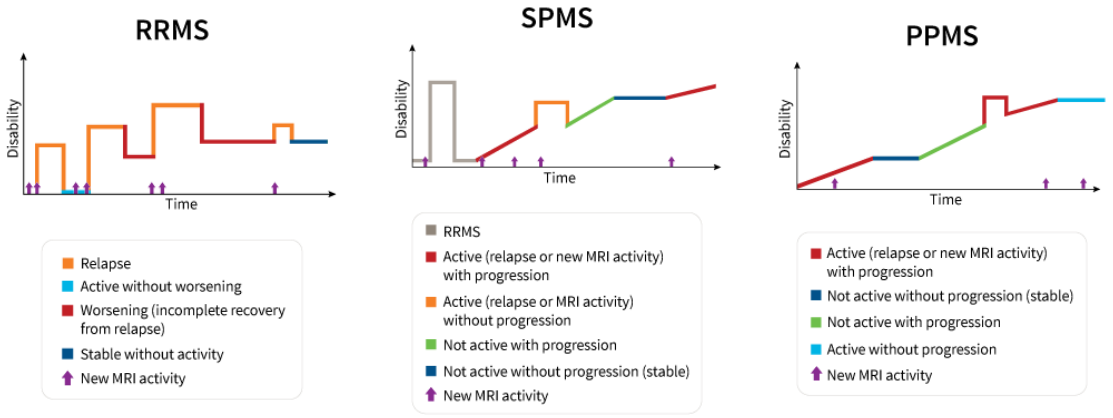
[6] two sample t-test
[7] chi-squared test for categorical data
[8] Number of patients that took antidepressant medication during the study

TASK-BASED FMRI. It is well know that MDD causes disruption in various cognitive domains, while the performance in working memory (WM) tasks is generally highly correlated with the loss of cognitive functions [63]. Neuroimaging studies based on functional Magnetic Resonance Imaging (fMRI) data acquired during tasks have become abundant in the past years. Indeed, the alteration observed in the blood-oxygen-level-dependent (BOLD) signal, has been found to be related to disruption in the cognitive ability of the subject. Our study is a *2-back WM task*: during the scanning session a sequence of stimuli is presented to the participant, who is asked to remember if the current stimuli matches the one observed two times back. In this study, stimuli were German nouns taken from the Berlin Affective Word List (BAWL; [185]) categorised as negative, positive and neutral. A *block design* was used during the experiment: each block consisted of 15 words shown in sequence and followed by a break (Fixation), for a total of 15 blocks, i.e. 5 per each type of stimuli. Then, a sequence of fMRI images were acquired on a Siemens Trio 3T (FUB) and a Philips Achieva 3T scanner (UZH) using standard echo planar imaging sequences [75, 156]. Standard preprocessing pipelines employing SPM [9] [141] were used, including mean registration, motion correction, and spatial smoothing.

RESTING STATE FMRI. Resting-state MRI data were also acquired on the same cohort at UZH and CHAR using the same scanners and similar parameters as for the task-based fMRI (see also Appendix A for details). The preprocessing pipeline was performed in Matlab (Version R2015a) using SPM and the CONN [10] toolbox (Version 17c; [193]) and consisted of the following steps: motion correction (realignment and unwarping), slice-timing correction, automatic detection of MRI artifacts, normalization to MNI space, and spatial smoothing (8 mm).

STRUCTURAL MRI. For the structural MRI images T1–weighted sequence (3D; magnetization-prepared rapid gradient echo) with an isotropic spatial resolution of 1 mm^3 was used, to allow a good differentiation between grey matter (GM), white matter (WM), and cerebrospinal fluid (CSF). Data were acquired at UZH and FUB, on the same scanner and cohort as for the resting state and task-based fMRI modalities. Head motion was reduced during scanning using a foam restraint. Preprocessing of the structural data was conducted using the default parameters from the CAT12 [11] toolbox [67] implemented in SPM. The T1–weighted images were corrected for bias field inhomogeneities, segmented into GM, WM and CSF [7] and spatially normalized using the DARTEL algorithm [6].

Figure 3.1: Multiple sclerosis disease subtypes [116]. Source: https://www.nationalmssociety.org/What-is-MS/Types-of-MS



3.1.2 Multiple sclerosis study

Multiple sclerosis (MS) is a chronic autoimmune, inflammatory neurological disease of the central nervous system (CNS), which targets and destroys the myelin and the axons [71]. The cause of MS is yet unknown, though it is generally accepted that it involves a combination of genetic and non-genetic factors, environmental or metabolic. Furthermore, the progression of MS is extremely varied and difficult to predict. Diagnoses is performed via a combination of clinical findings and assessments, such as walk impairment and the occurrence of at least 2 MS episodes. Additionally, the presence of lesions in the axons may be examined with the help of diagnostic tools, such as MRI images. Finally, inflammation of the CNS is also taken into account, as determined by the analysis of the CSF. Depending on the course of the disease, and in particular considering the frequency and severity of the episodes, MS patients can be categorised into four subgroups.

1. **Relapsing-remitting MS (RRMS).** It is the most common form of MS (approximately 80% − 85% of the patients are initially diagnosed with RRMS) and is characterised by an alternate sequence of relapsing episodes (attacks) and remission periods. During the remission periods, symptoms may completely disappear, partially continue, or become permanent. Following a relapse, the disability can either increase (worsening of the disease) and then stabilized (during remission), or get back to a state prior to the current episode.

2. **Secondary progressive MS (SPMS).** This status may develop in some patients initially diagnosed with RRMS. It is characterised by a continuous worsening of

[9] https://www.fil.ion.ucl.ac.uk/spm/software/spm12/
[10] https://web.conn-toolbox.org/
[11] http://www.neuro.uni-jena.de/cat/

the disease, and eventually of symptoms severity, with our without periods of remission.

3. **Primary progressive MS (PPMS).** Approximately 10% of MS patients belong this subgroup. Since the initial diagnosis, symptoms worsen over time and there are no relapses or remissions. Nonetheless, there might be periods of stable progression of disease severity, so–called plateau.

4. **Progressive-relapsing MS (PRMS).** PRMS is the rarest form of MS and 5% of patients are diagnosed with it. Progression of symptoms occurs from the start, with alternate worsening and no remission periods.

Despite these MS subgroups being clinically established, it is often controversial to assign patients into the correct subcategory. A subject may also be placed into a different category over time and the characterisation into a MS subtype is usually done retrospectively, since at the initial phase of the disease is yet unclear how the course will evolve. Besides, in recent literature PRMS has been disregarded as MS category and those subjects are now categorised as PPMS [116]. For the purpose of this thesis, we will use this characterisation. An overview of the MS disease course with these latest three subtypes is depicted in Figure 3.1.

Data selection

The data sample for the MS study include 12 HC, as well as 13 patients with PPMS, 26 patients with RRMS, and 18 patients with SPMS. The subjects were recruited at the University College London (UCL) with MRI scanned acquired at baseline and after 24 months (approximately). The clinical assessment was performed by an expert and the severity of the disease was evaluated via the Expanded Disability Status Scale (EDSS; [108]). The subjects were then grouped into their corresponding subtype for the retrospective analysis.

Data acquisition and preprocessing

All MRI images were acquires on a 3T Philips Achieva scan for multiple modalities: conventional structural imaging (volumetric 3D T1-weighted gradient echo imaging; 2D PD-weighted and T2-weighted axial spin echo imaging); single-shell diffusion-weighted magnetic resonance imaging (DWI); magnetization transfer (MT)–weighted imaging. Standard image preprocessing was performed consistently within the cohort using the FMRIB Software Library [12] (FSL; [94]). For diffusion images, the preprocessing steps included correction for motion and eddy currents (FSL eddy) and for Echo-planar imaging (EPI) distortions (BrainSuite [13]; [157]). Affine co-registration to a mid-way MRI space obtained from baseline and follow up diffusion images was applied between all the modalities: MT, anatomical images, and diffusion. From the MT–weighted and

[12] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL
[13] http://brainsuite.org/

DWI images, magnetization transfer ratio (MTR) map and diffusion tensor imaging (DTI) metrics were also calculated in voxel-by-voxel fashion (see Section 3.3.2). Additional information on the data modalities, acquisition, and preprocessing is given in Appendix A.

3.2 Pattern analysis methods in neuroimaging

In this section, we present the core methodology and pipelines developed for the imaging studies. We first introduce classical univariate analysis approaches (statistical parametric map (SPM); Section 3.2.1) and discuss their limitations, leading us to the extension to multivariate methods which we will present in Section 3.2.2. We conclude by illustrating the Multiple Kernel Learning (MKL) approach, expanding the multi-variate analysis from uni-modal to multi-modal images (Section 3.2.3).

3.2.1 Univariate analysis: statistical parametric map

Univariate mass voxel-based analysis methods introduced by Friston *et al.* [55] have been widely used to analyse fMRI data, with the scope to obtain a map of the brain area activated under a given condition [37]. This approach relies on a general linear model (GLM) to extract statistical parametric maps (SPM), which provides information on the association of individual voxels with respect to a certain hypothesis. For example, in the WM block-design fMRI task, we might be interested to determine group of voxels that activate with a given condition (e.g. positive, negative, or neutral stimuli), for the group of MDD or HC. The classical GLM analysis consists of two major steps:

1. First-level analysis: within subject;
2. Second-level analysis: across subjects.

General linear model: first–level analysis

The *first-level analysis* models the time series fMRI for each voxel and subject independently, inferring the relationship between the BOLD fMRI signal over time as a function of the experimental design. For the moment, let us assume to have a sequence of scans for a single individual. We denote by $y_i \in \mathbb{R}^T$ the signal of a single voxel v_i over time, where T corresponds to the number of 3D images acquired per each subject, i.e. the length of the time series. We define the design matrix $X \in \mathbb{R}^{n \times p}$ encoding the experimental design information (for instance a block matrix for a block design experiment) and potential confounding factors; here, n is the number of voxels per scan and p depends on the experiment and covariates. The first–level GLM establishes a simple linear dependency between the signal of a single voxel and the experimental design matrix:

$$y_i = X\beta + \epsilon. \tag{3.1}$$

In agreement with the classical linear regression model, the errors are assumed to be normally distributed $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, and ordinary least squares is employed to solve the problem independently for each voxel $y_i,\ i = 1, \ldots, n$. The parameters

$$\beta = \beta_{i,1}, \ldots, \beta_{i,p}$$

for each voxel v_i are estimated from the GLM model. Aggregating the beta values over the voxel space results in a set of 3D

$$\textit{beta images}: \ \beta_1, \ldots, \beta_p$$

for a single-subject and for each experimental condition. The beta images are directly related to the activation level of the subject with respect to the experimental design, for instance a particular stimuli. To represent the difference between experimental conditions, one can derive *contrast images* from the beta parameters. These are linear combinations, with the weights defining the relation of interest. Formally, the contrasts are determined by *contrast weights* and can be written as:

$$c = [c_1, \ldots, c_p] \tag{3.2}$$
$$C = c\beta. \tag{3.3}$$

For example, one can choose $c_0 = 1$ and $c_1 = -1$ to obtain the contrast C to find voxels that are more active in condition 0 (β_0) than condition 1 (β_1). As for the beta images, the contrasts are defined as $c_{i,1}, \ldots, c_{i,q}$ for each voxel v_i and number of combinations of interest q. The contrast images also are MRI images themselves.

GLM SECOND-LEVEL ANALYSIS.

Nevertheless, to analyse the contrast images and understand the different reactions of the brain in group of subjects, we need to take an extra step. The *second-level analysis* extends the first-level mass univariate approach to the group level. Considering a cohort with subjects belonging to two (or more) different groups, such as patients and controls, we are interested to tackle brain areas that are showing a consistent activation behaviour within the group. Then, the statistical analysis assesses the difference across activation at the group level. In other words, we might want to know whether a contrast depicts a peculiar activation pattern (group of voxels higher or lower activated) across groups of subjects. A $t-test$ is used for this scope, after averaging the contrasts within the group.

LIMITATIONS OF THE STATISTICAL ANALYSIS

While being a classical and well established procedure, the standard GLM exhibits several pitfalls. To begin with, since the first-level GLM treats the voxels independently, their interactions are not taken into account. Secondly, the mass-univariate GLM is a applied to a very large voxel space. Typically, an fMRI image contains more than 100000

voxels resulting in a huge multiple comparison problem: for example, assuming that 100000 is the voxels dimension, choosing a significance level $\alpha = 0.005$ will potentially result in 5000 voxels to be significant by chance (false positives). To overcome this problem, methods for multiple comparison correction may be employed [131]. However, it is not straightforward to decide about a meaningful, yet not too conservative approach and there is no general rule on how this threshold should be defined [46, 171]. In the last decade, the advent of machine learning methods has revolutionised a lot of fields. Then it should not come as a surprise that these methods have also been extended to the neuroimaging area to conquer its limitations, leading to several methodological and experimental improvements.

3.2.2 Multivariate classification analysis for brain imaging data

Multi-variate analysis approaches rely on state-of-the-art supervised or unsupervised machine learning techniques and have been initially proposed to overcome the two major limitations of the GLM approaches: (1) the missing interaction across voxels; (2) the multiple comparison problem. In this section, we will mostly focus on the multi-voxel pattern classification (MVPC) method which combines machine learning classifiers and feature selection methods for the analysis of imaging data. We will integrate our description with regression and clustering based techniques in Chapter 4.

The goal of MVPC is twofold. On one hand, we are interested in finding patterns of activities that differ across experimental conditions and groups of subjects. On the other hand, the classification component aims at predicting the group label of a new unseen out-of-sample image with maximal accuracy, based on a model learned on the available training data. More precisely, given a standard classification problem, we want to learn the decision function f to predict $f(x) = y \in 0, 1$ for a new input data $x \in \mathbb{R}^p$, where f has been learned on the available training data. In our case, $x \in \mathbb{R}^p$ is an MRI image and the goal is to find a proper trade-off to learn a model to jointly maximise prediction accuracy and getting interpretable patterns. When the relationship between the outcome y and features x is linear, the pattern interpretation can be inferred from the weights of the learned model directly: this is the case with linear support vector machines (SVMs) [23, 35]. We briefly recall the standard C-SVM model, which can be roughly described as finding the optimal hyperplane separating two classes, while maximising the margin in the training data:

$$\begin{aligned} &min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t. } y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i,\ \xi_i > 0,\ \forall i = 0, \ldots, n-1. \end{aligned} \tag{3.4}$$

In Equation 3.4, $x_i \in \mathbb{R}^p$ is the feature vector of sample i, $w \in \mathbb{R}^p$ is the weight vector, while b is the bias term; the regularization parameter C determines the misclassification rate, and ξ_i provides an upper bound for the number of training errors. Assuming that the feature space x coincides with the imaging voxel space, we can interpret the learned weights as indicating the relevance of that brain area (or voxel) with respect to

the classification task. From now on, we will refer to Equation 3.4 as the SVM or C-SVM model, indistinguishably.

In multiple machine learning applications and especially in neuroimaging studies, a critical issue is given by the *curse of dimensionality*, where the number of samples is much lower than the number of features $n \ll p$. This setting can generate several issues, including overfitting and computational burden [14]. To partially leverage for it, the SVM can be equipped with a feature selection or dimensionality reduction strategy and paired with a leave-one-out cross validation [178, 179]. Any feature selection strategy can be easy combined with SVMs [34]. We choose the *F-score* criteria, a computationally fast and meaningful way to assess the correlation between each feature and the classification task. The *F-score* method is a univariate feature selection strategy that ranks their importance based on the *F-value*:

$$F = \frac{\text{between-groups variance}}{\text{within-group variance}} = \frac{\sum_{k=1}^{K} n_k (\sum_{j=1}^{n_k} x_{k,j} - \sum_{i=1}^{n} x_i)/(K-1)}{\sum_{k=1}^{K} \sum_{j=1}^{n_k} (x_{k,j} - \frac{1}{n_k} \sum_{j=1}^{n_k} x_{k,j})^2/(n-K)}. \tag{3.5}$$

Here, n is the total number of sample; K is the number of classes; n_k is the number of samples in class k. We will denote the method that performs feature selection via *F-value* combined with the C-SVM classifier as *SVM-fScore*.

While a linear classification method is very convenient in terms of interpretability, it is possible that other classifiers have an higher discriminative power and result in better classification performance. In principle, we can extend the MVPC pipeline with any classifier that can take as input the flatten images in a vectorial representation. Furthermore, the kernel trick (Section 2.1) allows to apply different non-linear kernels on data, to input directly in the classifier. This is achieved by reformulating Equation 3.4, as we will expand in the next Section 3.2.3. Other popular classifiers include RandomForest and k-NearestNeighbours. However, neither of them provide an easy interpretation of the selected features. Furthermore, MVPC can also be equipped with different feature selection methods to replace the *F-score* approach. A widely popular technique in both genetic and medical imaging studies is *recursive feature elimination* (RFE) [77], which is based on an iterative selection of the features that minimise the generalisation error with respect to the classification task. Since this is an *embedded method*, the learning performance is usually very good, though it has the disadvantage of having to re-fit the SVM model at each iteration, thus being computationally expensive. We will refer to the SVM method equipped with recursive feature elimination as *SVM-RFE*. For a detailed overview and introduction to classification strategies and feature selection methods we refer the interested reader to Bishop [16] and Friedman *et al.* [54].

3.2.3 Multi-modal analysis

The methods discussed so far are designed to work on single input feature space. Nonetheless, in many clinical settings it is common to acquire more than one image modality per subject, during a single or multiple scanning sessions. Different modalities carry unique information about certain lesions or aspects of the brain which may be

affected in a specific task (see Appendix A for an overview of MRI modalities). Therefore, methods to analyse and efficiently integrate the relevant information from multiple images have been developed as a natural follow-up to the multi-variate unimodal approach. While it is possible to consider every modality as independent from each other, examining their interaction has the potential to drastically improve not only the classification performance but also the clinical practice. For instance, a clinician provided with a quantitative measure about the relevance of acquiring a given modality for a task of interest, would be guided on the acquisition protocol for the patient by the automated healthcare system.

The easiest way to combine different source of data is via feature concatenation. Trivially, given two samples $x_{p_1} \in \mathbb{R}^{n \times p_1}$ and $x_{p_2} \in \mathbb{R}^{n \times p_2}$, we define the joint sample as $x_{p_1+p_2} = (x_{p_1}, x_{p_2}) \in \mathbb{R}^{n \times (p_1+p_2)}$. The same algorithms can be applied on the new concatenated instances to infer the result on the combined data. Another classical strategy to combine modalities is the so–called *ensemble learning*. The high level idea is to run the model separately on each of the data source, e.g. x_{p_1} and x_{p_2}, and aggregate the performance of the model in a post-hoc fashion, for example by averaging the results. However, none of these strategies explicitly take into account the interaction between data sources. In Section 2.1 we discussed how kernels provide a measure of similarity between objects in an implicit form, while different kernels detect specific relations in the original feature space. Using the closure properties, it is possible to apply and combine different kind of kernels on the same object and features, obtaining a joint similarity measure.

As before, let $x_{p_1} = x_1, x_2, \ldots, x_n \in \mathbb{R}^{p_1}$ be n objects in the same feature space and denote the corresponding kernel matrices as $k_{RBF}(x_{p_1})$ and $k_{lin}(x_{p_1})$. One can define the combined kernel matrix as

$$k(x_{p_1}) = k_{RBF}(x_{p_1}) + k_{lin}(X_{p_1}), \tag{3.6}$$

representing a joint similarity value of the linear and RBF kernels. Similarly, let $x_{p_2} = x_1, x_2, \ldots, x_n \in \mathbb{R}^{p_2}$ be the same n objects in a different feature space. Then we can define a linear kernel on this feature space as $k_{lin}(x_{p_2})$ to obtain a joint kernel on x_{p1+p2} as

$$k(x_{p1+p2}) = k_{lin}(x_{p_1}) + k_{lin}(x_{p_2}), \tag{3.7}$$

combining the information from the two spaces. We used the sum for simplicity, but any weighted linear combination could be applied. The family of methods dealing with the analysis, optimisation and generation of these combined kernel is called Multiple Kernel Learning (MKL) [72]. We will now provide an overview of these methods, with particular emphasis to their application in neuroimaging and for our data cohort.

Multiple Kernel Learning

To combine multiple kernel matrices and encode them as joint input for a classifier or regressor, the first step is to "kernelise" the SVM to allow an input kernel matrix in

the learning algorithm. This is achieved by applying the Lagrangian dual function in Equation 3.4 and solve the alternative optimisation problem:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \\ &\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq C, \ \forall i = 1, \ldots, n \end{aligned} \tag{3.8}$$

where α is the vector of Lagrangian dual variables. The inner product $\langle \phi(x_i), \phi(x_j) \rangle$ is a kernel between x_i and x_j, leading to the predicting model being expressed as:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b. \tag{3.9}$$

From the closure properties of kernels, it follows that multiple linear combinations of kernels can be defined and still be a valid instance to employ in Equation 3.8. Different optimisation strategies to combine the kernels result in different solutions. We consider a general combination of the form

$$k_{MKL} = \sum_{r=1}^{R} \beta_r k_r, \tag{3.10}$$

where R is the number of kernels to aggregate. The optimization problem consists in finding the best parameter values β_r such that k_{MKL} is mostly representative of the joint similarity. We further aim to maximise the predictive power of the model (Equation 3.9) with respect to the generalization error of the classification or regression problem. The simple choce of $\beta_r = \frac{1}{R}$ and $r = 1$, for each $r = 1, \ldots, R$, leads to the average and sum of kernels, respectively. We will refer to the average MKL approach as *avgMKL*. The ambition of any MKL algorithm is to outperform the average and sum baselines, although these simple combinations often exhibit a good empirical performance. Besides, they have the advantage of being computationally cheap, given that no tuning strategy is required to find the optimal β_r.

Easy Multiple Kernel Learning

We now present the particular instance of MKL employed in our experiments [1]. An overview of different approaches can be found in Gönen and Alpaydin [72].

Assume we are given a set of training samples $\{x_1, \ldots, x_n\}$ with $x_i \in \mathbb{R}^p$ and associated class labels $y_i = \{1, -1\}$, for $i = 1, \ldots, n$. We denote the arbitrary kernel matrix as $K \in \mathbb{R}^{n \times n}$ and let $y = \{y_1, \ldots, y_n\}$ be the set of all class labels. The subset of positive and negative training samples are denoted as y^+, y^- where $i \in y^+, y_i = 1$ and

$i \in y^-, y_i = -1$, respectively. Then, we consider the set of corresponding probability distribution, namely:

$$\Gamma = \{\gamma > 0 : \sum_{i \in y^+} \gamma_{i=1}, \sum_{i \in y^-} \gamma_i = 1\}. \tag{3.11}$$

Aiolli and Donini [1] propose to exploit the domain of Γ distributions and solve the MKL problem via a the KOMD (Kernel method for Optimization of the Margin; [2]) algorithm. More precisely, the data distribution of the training class labels is included to constrain the γ vector to better generalise on unseen data, that is:

$$\min_{\gamma \in \Gamma} D(\gamma) := \gamma^T y K y \gamma. \tag{3.12}$$

Defining $R(\gamma) = \gamma^T \gamma$ the final optimization problem is reformulated as:

$$\min_{\gamma \in \Gamma} (1 - \lambda) D(\gamma) + \lambda R(\gamma), \tag{3.13}$$

where $\lambda \in [0, 1]$ plays the role of a regularization parameter, encouraging a low variance solution via the term $R(\gamma)$. The prediction function for a new example x_{new} is given by:

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} \gamma_i y_i k(x_i, x_{new}) = k_{new}(x) y \gamma, \\ k_{new}(x) &= [k(x_i, x_{new}), \ldots, k(x_n, x_{new})]^T. \end{aligned} \tag{3.14}$$

We now plug-in the MKL component by defining $k = \sum_{\beta=1}^{R} \beta_r k_r$, $\beta_r \geq 0$ and reducing to solve Equation 3.13 with respect to γ and β, simultaneously. Ultimately, we want to maximise the distance between the positive and negative samples, i.e.:

$$\max_{\|\beta\|=1} \min_{\gamma \in \Gamma} (1 - \lambda) \gamma^T y (\sum_{r=1}^{R} \beta_r K_r) y \gamma + \lambda \|\gamma\|^2. \tag{3.15}$$

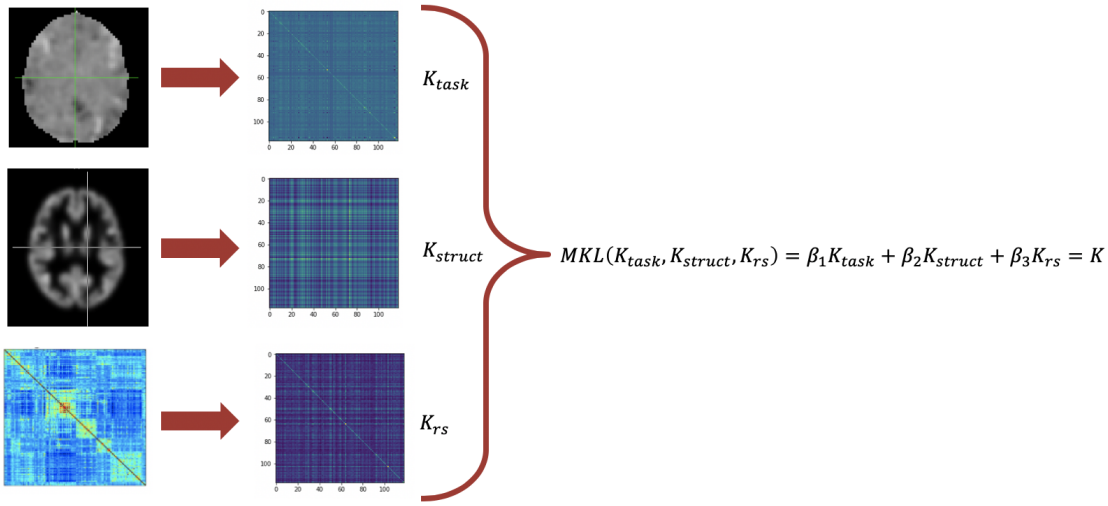
Aiolli and Donini [1] show that by rewriting $D_\beta(\gamma) = \{\gamma^T y k_1 Y \gamma, \ldots, \gamma^T y K_R Y \gamma\}$, there exist an analytic solution $\beta^\star = \frac{D_\beta(\gamma)}{\|D_\beta(\gamma)\|}$ of Equation 3.15. Plugging in $\beta^\star$ we obtain:

$$\min_{\gamma \in \Gamma} Q(\beta^\star, \gamma) := (1 - \lambda) \|D_\beta(\gamma)\| + \lambda \|\gamma\|^2. \tag{3.16}$$

It follows that the optimization of the MKL problem $Q(\beta^\star, \gamma)$ is equivalent to solve the KOMD algorithm for a single kernel matrix (see Equation 3.13). We refer to this method as *EasyMKL*. In practice, an upper bound of Equation 3.16 is minimized; details on the optimization strategy can be found in Aiolli *et al.* [2].

For our scope, MKL is applied on the MRI data by defining a single kernel per modality, assuming that for a subject a set of multi-modal images has been acquired. We will

Figure 3.3: Multiple Kernel Learning pipeline on the MDD data. Linear combination of kernels defined on the three modalities: sMRI, task-based fMRI and resting-state fMRI.



elaborate on this in the next sections; an overview of the MKL strategy on the MDD study with three modalities is shown in Figure 3.3.

3.3 Feature extraction

All the algorithms described so far require a meaningful set of features extracted from the image, which are informative of their characteristics, and a suitable input for the classifiers. At an high level, we distinguish between two major feature extraction approaches:

(i) High-dimensional high-resolution whole brain features

(ii) Low-dimensional low-resolution region of interest (ROI) features

In (i) the feature space is inferred from the whole brain in a voxel-by-voxel fashion, leading to a very high dimensional representation containing single voxel signal. In (ii), the focus is shifted on specific brain areas or regions of interest (ROI), either in a voxel-by-voxel fashion or by summarising the information within a region, resulting in a significant reduction of the dimensionality. In either case, the feature extraction can be coupled with classical feature selection or dimensionality reduction techniques (Principal Component Analysis (PCA); Independent Components Analysis (ICA); [54]) to further reduce the search space. This step is particularly relevant in neuroimaging studies, due to the sample size being generally much smaller than the number of features. More recently, deep learning based methods have become incredibly popular in the neuroimaging community, and applied as feature extractor either in an unsupervised or supervised fashion [3].

Table 3.2: Beta and contrast images for the WM block design experiment

(a) Beta parameters

Beta image ID	Condition
Beta01	negative stimuli
Beta02	neutral stimuli
Beta03	positive stimuli
Beta04	break stimuli
Beta05 - Beta10	movement parameter
Beta11	constant

(b) Contrasts of interest.

Contrast ID (#)	Contrast Type
Con01	WM > Fix
Con02	Pos > Fix
Con03	Neg > Fix
Con04	Neu > Fix
Con05	Emo > Neu

3.3.1 High-dimensional features

For the MDD study, we extract voxel–by–voxel high dimensional features either directly from the image or after some preprocessing step to select the most relevant information. For each of the modality described in Section 3.1.1, a different strategy is considered.

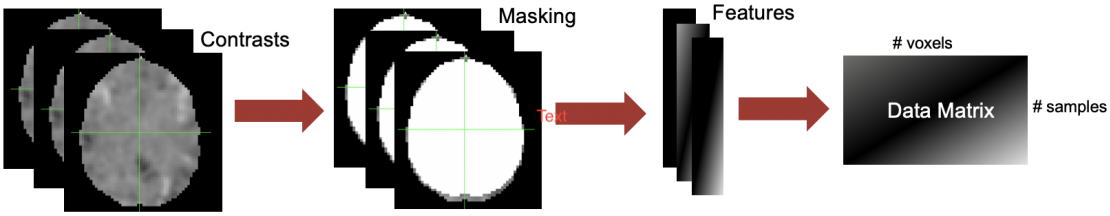
Task-based fMRI

We observed that the the first-level analysis of the classical GLM ignores the group and interaction components of the data, only focusing on single subject analysis. On the preprocessed task–based fMRI, a single subject general linear model analysis was performed. The haemodynamic response (HR) as was modelled as explanatory variable, while the different conditions (Fixation, Negative, Neutral, Positive) and realignment parameters were included as independent variables in a block design [55]. As an outcome, contrast images for each subject for the following contrasts were derived, as summarised in Table 3.2:

1. All WM conditions versus fixation condition (WM > Fixation);
2. Positive WM condition versus fixation condition (Pos > Fixation);
3. Negative WM condition versus fixation condition (Neg > Fixation);
4. Neutral WM condition versus fixation condition (Neu > Fixation);
5. Emotional (positive and negative) WM conditions versus neutral WM condition (Emo > Neutral).

We use the *beta* and *contrast* images obtained from the first level analysis to extract features for the task–based fMRI. To extract voxel–wise brain tissue and disregard the background noise, we computed a group level mask for each of the *beta* and *contrast* image. The mask has been computed using a 90% threshold at a group level, meaning that a voxel was included in the group mask if and only if it was identified as brain

Figure 3.4: Machine learning pipeline on task-based fMRI for the MDD study



tissue in at least 90% of the subjects. In a second step, we flattened the voxel space into a one dimensional array per subject and per image type. In this context, *beta* and *contrast* images are informative of the activation voxels at a particular block in the experimental design or of difference in signal between experimental tasks. For each beta and contrast image, we finally obtain a data matrix $X_{task} \in \mathbb{R}^{118\times41248}$ as input for the classifier. The pipeline is depicted in Figure 3.4.

Resting state MRI

On the resting state data, the CONN toolbox was used to obtain pairwise correlation measures for each region. More precisely, 106 regions from the FSL Harvard-Oxford Atlas were obtained to calculate ROI-to-ROI correlation maps, by leveraging the residual blood oxygen level-dependent (BOLD) time courses between pairs of regions and computing Pearson's correlation coefficients. The correlation coefficient were converted to normally distributed z-scores using the Fisher transformation to improve the validity of second-level General Linear Model analysis [64]. The correlation matrices obtained by this procedure are treated as features in the machine learning pipeline, by using their lower/upper diagonal, resulting in a data matrix $X_{rs} \in \mathbb{R}^{118\times5565}$.

Structural MRI

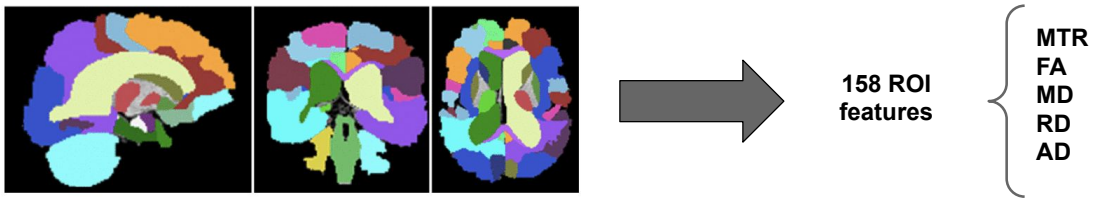
We further extract sMRI based features from the whole–brain T1 images by masking the brain tissue with an analogous procedure as for the task–based fMRI images. Of course, the ultimate feature space dimension will change given the differences in resolution and group mask generation between the task–based and sMRI data. This procedure results in a resting–state data matrix $X_{sruct} \in \mathbb{R}^{118\times483591}$.

3.3.2 Low resolution region of interest features

In the MS cohort, we begin by extracting a set of imaging based features obtained from diffusion and MT images, as a result of whole–brain approach and segmentation preprocessing pipelines. The following metrics are derived:

1. White matter MS lesions maps extracted manually by an expert neurologist on the PD-weigthed/T2/weighted scans;

Figure 3.5: ROI parcellation with GIF.



2. Cortical thickness calculated with GIF and FreeSurfer [14] [150];
3. MTR map calculated from the MT-weighted scans voxel–by–voxel;
4. DTI metrics, calculated from DWI images voxel-by-voxel: Fractional Anisotropy (FA), Mean Diffusivity (MD), Radial Diffusivity (RD), Axial Diffusivity (AD).

We focus on (3) and (4): here the voxel–by–voxel metrics are also images obtained from the preprocessed DWI and MT scans, respectively. Contrarily to the whole–brain approach, we perform a ROI based feature extraction on these data. We use a brain parcellation strategy to select ROIs obtained via GIF algorithms [15] [28, 144] which results in 158 regions. We subsequently average the voxel values within each ROI to get a single value measure of the selected brain area. Therefore, for every metric (image) we obtain a data matrix in the low-dimensional ROI space: $X_{mt} \in \mathbb{R}^{69\times158}$; $X_{fa} \in \mathbb{R}^{69\times158}$; $X_{md} \in \mathbb{R}^{69\times158}$; $X_{rd} \in \mathbb{R}^{69\times158}$; $X_{ad} \in \mathbb{R}^{69\times158}$. The subscripts stand for the respective metric. An overview of the feature extraction strategy is depicted in Figure 3.5.

3.4 Experiments

We perform an extensive experimental study to analyse and investigate the properties of the MDD and MS studies with respect to various prediction tasks. We compare the classification performance of different machine learning models on a patient vs control problem and assess the individual contribution of each modality, as well as of their combination. The following tasks are considered in our experimental setting:

(i) *DPvsHC - uni.* The data matrices X_{struct}, X_{task} and X_{rs} are all used independently as input for the model classifier. The X_{task} matrix indicates a general task-based MRI image. However, the different *beta* and *contrast* images can be handled separately, therefore we index the corresponding data matrix by the beta or contrast ID as reported in Table 3.2. For example: $X_{\beta02}$ denotes the data matrix of MRI *beta* image of the *neutral condition*; X_{c01} denotes the data matrix from the MRI image of the *WM > Break* contrast. When used, the corresponding kernel matrices are

[14] https://surfer.nmr.mgh.harvard.edu/
[15] http://niftyweb.cs.ucl.ac.uk/

also similarly indexed, for instance $K_{lin,\beta 02}$ denotes the kernel matrix obtained via a linear kernel on $X_{\beta 02}$.

(ii) *DPvsHC - multi.* In the multi-modal scenario, the data or kernel matrices of the various modalities are combined and optimised, with the goal to pick the most relevant information from each of them. We use a "+" symbol to denote the combination of two or more images.

(iii) *MSvsHC - uni.* This scenario is comparable to (1), despite the low-dimensional ROI feature matrix is used from the respective metric of interest.

(iv) *MSvsHC - multi.* We combine metrics from (3) in an analogous fashion as for the DPvsHC - multi.

3.4.1 Experimental setup

We compare the classification performance of a variety of linear an non-linear classifiers. As we mentioned, the linear classifiers have the advantage of being easy to interpret since a one-to-one correspondence between the feature importance and prediction task can be derived. Nevertheless, it is often the case that non–linear interactions play a crucial role to build a predictive model and cannot be ignored. For the uni-modal analysis the following settings are used.

(i) *linSVM.* A support vector machine with linear kernel. The C parameter is chosen in the grid $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

(ii) *rbfSVM.* A support vector machine with gaussian kernel. The C parameter is chosen in the grid $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ and the γ parameter is chosen in the grid $\{2^{-3}, 2^{-2}, \ldots 2^{3}\}$.

(iii) *polySVM.* A support vector machine with polynomial kernel. The C parameter is chosen in the grid $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. We use wither degree 2 or 3.

(iv) *sigSVM.* A support vector machine with sigmoid kernel. The C parameter is chosen in the grid $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

(v) *LR.* A logistic regression classifier. The regularisation C parameter is chosen in the grid $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

(vi) *LDA.* A linear discriminant analysis classifier. No hyperparameters are tuned.

(vii) *KNN.* A k - nearest neighbours classifier. The parameter k determining the number of neighbours is chosen in the grid $\{2, 5, 7, 10\}$.

(viii) *RF.* A Random Forest classifier. The parameter *n-trees* determining the number of trees in the forest is chosen in the grid $\{10, 20, 30\}$.

We refer to the method equipped with the corresponding feature selection as * *-fScore* and * *-RFE*, where * is one of the methods mentioned above; if not denoted otherwise, we imply that no feature selection is performed. When selecting the features, we choose a range in the grid $\{10\%, 20\%, \ldots, 100\%\}$ of the original dimension. In the multi-modal setting the following methods are considered:

(i) *Feature Concatenation (FC):* a simple concatenation of the uni-modal features per subjects.

(ii) *Average Multiple Kernel Learning (avgMKL):* the MKL approach with uniform kernel weights, either with linear (linAVG) or RBF (rbfAVG) kernel. The regularisation C parameter is chosen in the grid $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ and the γ parameter for the RBF is chosen in the grid $\gamma = \{2^{-3}, 2^{-2}, \ldots 2^{3}\}$.

(iii) *EasyMKL:* the Easy MKL method introduced in Section 3.2.3, with λ parameter tuned in the range $\lambda = \{0.001, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$. As for avgMKL, the approach will be denoted as linEASY or rbfEASY depending on the kernel chosen; γ and C parameters are chosen in the same range as for avgMKL.

All the experiments are performed via a nested leave—one–out cross validation (LOOCV), with all the parameters selected on the training set only via 5 fold cross validation (CV).

Evaluation

In the context of clinical studies, it is extremely relevant to evaluate the classifier in terms of different criteria that can account for the medical relevance. For example, for a physician might be more important if the model prevents a large occurrence of false negatives, since diagnosing a sick patient as healthy is more risky than wrongly classify an healthy subject as sick. Therefore, we are especially interested in considering evaluation criteria that can take this risk assessment specifically into account. The following abbreviations are used:

- TP: true positives. Number of patients correctly classified as such.
- TN: true negatives. Number of healthy controls correctly classified as such.
- FP: false positives. Number of healthy controls wrongly classified as patients.
- FN: false negatives. Number of patients wrongly classified as healthy controls.

Using these four counting characterisation of the predicted classes, we define a set of well established evaluation metrics as follow:

- Accuracy $= \frac{TP+TN}{TP+FP+TN+FN}$
- Precision $= \frac{TP}{TP+FP}$
- Recall/Sensitivity $= \frac{TP}{TP+FN}$

- Specificity $= \frac{TN}{TN+FP}$

The *sensitivity* is particularly relevant in the medical context, assessing how many of the patients (TP+FN) have been correctly identified. Similarly, the specificity measures the proportion of healthy individuals (TN+FP) that are effectively healthy (TN). An important metric taking both these measures into account idthe receiver operation characteristic curve (ROC-curve), which shows the true positive rate (TPR; sensitivity, recall) versus the false positive rate (FPR; 1-specificity). Additionally, the Precision-Recall curve (ROC-PR-curve) is also informative of the relationship between the respective quantities, with the advantage of being quite robust to unbalanced data.

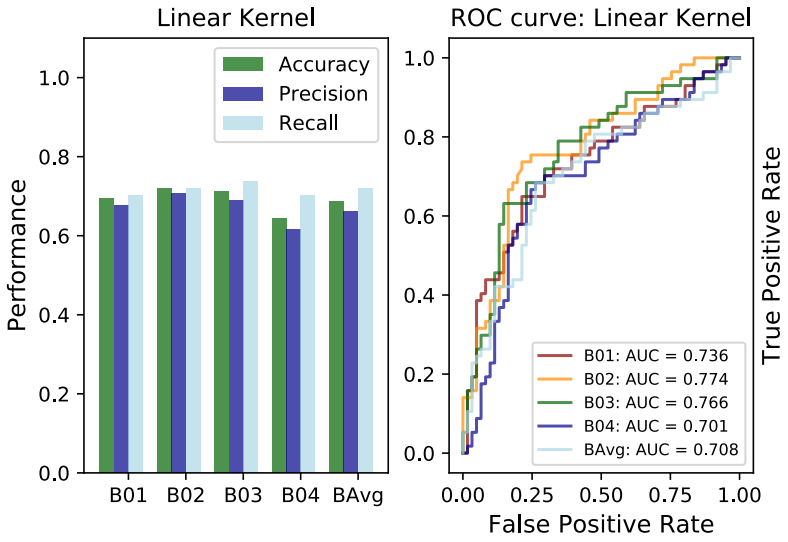
PERMUTATION TEST. To further corroborate the predictive power of our classifier, a permutation test can be used to assess the statistical significance of the results. The predicted labels are permuted 1000 times and the whole machine learning pipeline is re-applied considering the permuted labels as the real ones. The null hypothesis states that the group predicted labels are randomly assigned; a *p-value* is calculated by assessing the probability of observing a predictive performance better or equal than the calculated one.

POST-PROCESSING ANALYSIS

INTERPRETABILITY. In the uni-modal linear classification scenario, we can interpret the weights obtained from the SVM as proportional to the corresponding feature discriminative power in the classifiers. In practice, in our applied label convention denoting the positive class (patients) by $y = +1$ and the negative class by $y = -1$, a large positive weight depicts more activation in that voxels of the patients, while a large negative weight is a signal of more activation in the healthy controls [60, 81]. Nevertheless, the nested CV approach implies a possibly different set of features, and consequently weight maps, in each fold. To obtain the final unique weight map different aggregation strategies can be employed; in our experiments, we use a simple average and select the number of features via majority vote across folds [179].

POST-HOC ROI. The weight maps obtained from the SVM do not account for the spatial distribution of the voxels within the image, consequently it is possible to obtain isolated voxels with high values. While from a methodological perspective this is acceptable, in the neuroimaging context we are interested to find patterns as cluster of voxels that activate together, as representative of one or more brain areas. To create post-hoc ROI clusters we kept 20% of the highest classification weights, masked the remaining, and used a cluster threshold of 50 voxels at the minimum, to overcome obtaining small and isolated activation area. The difference between groups was assessed via an analysis of variance (ANOVA) and a Bonferroni correction was used to account for multiple testing.

Figure 3.6: Performance of multiple beta images with a linear SVM. The left plot shows accuracy precision and recall for each beta and for the average image. On the right side, the ROC curve is reported.



EFFECT OF SCANNER TYPE. In medical imaging, the type of scanner used and its setup is a key confounding element in any statistical or machine learning based analysis. This is due to the intrinsic properties of each scanner that generate unique images, implying that a naive model that does not take into account the scanner type might pick up a signal that is informative of the acquisition site, rather than of the clinical group. While this is generally implicitly handled in automated machine learning model, investigating the effect of the scanner type in a post-hoc study provides an extra validation layer to the analysis. Our assessment includes three scenarios that explicitly consider the Zurich or Berlin acquisition site as a label for the data:

(i) classification performed on the entire sample and evaluation on single site data, i.e. the performance is separately evaluated on the subsets of Zurich and Berlin;

(ii) classification performed on the subgroups of Zurich and Berlin data, separately and independently;

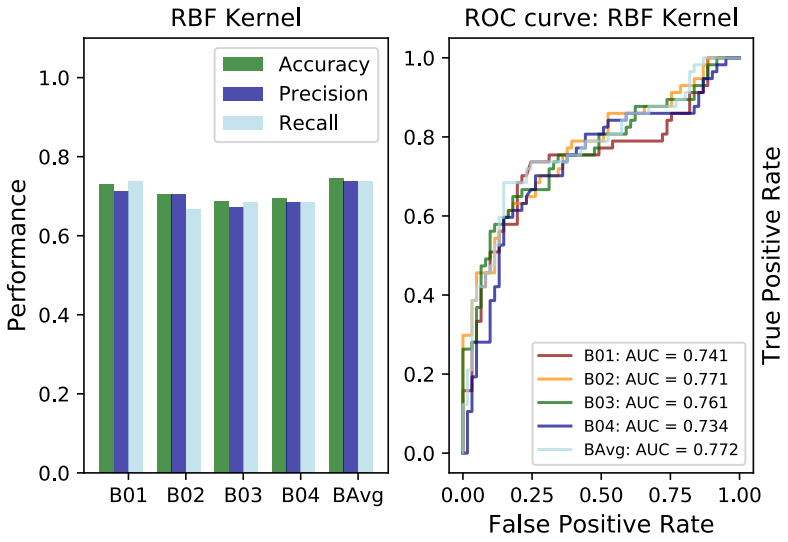
(iii) classification on the Zurich data by training on the Berlin data only, that is a single train/test split with Berlin/Zurich, respectively.

3.4.2 RESULTS

MAJOR DEPRESSIVE DISORDER STUDY

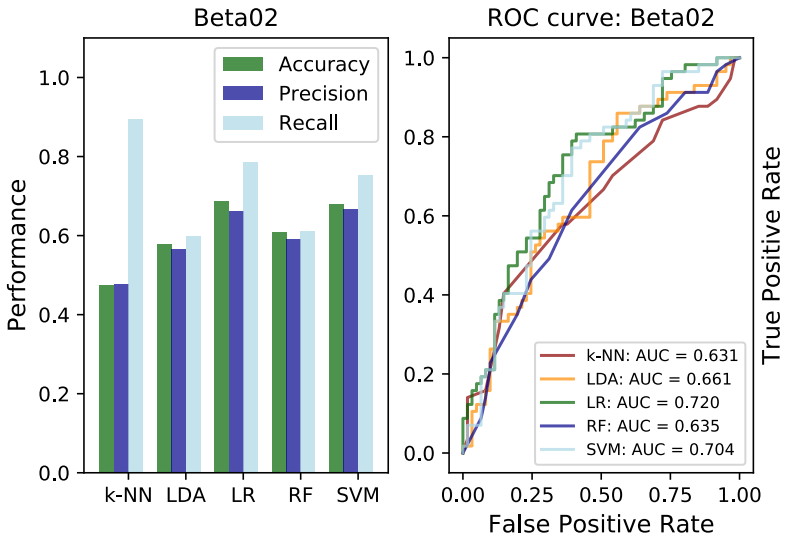
UNI-MODAL CLASSIFICATION. We begin by examining the *DPvsHC - uni* task for the MDD study, evaluating the classification performance of the different classifiers on each

Figure 3.7: Performance of multiple beta images with an rbfSVM. The left plot shows accuracy precision and recall for each beta and for the average image. On the right side, the ROC curve is reported.



of the beta images. In Figure 3.6 and Figure 3.7 we report the results with a linear and RBF kernel, on the individual beta images and their average. Additional results with other kernels are reported in Appendix A. We observe that the beta02 data matrix, corresponding to the neutral stimuli condition, achieves the best performance on both the RBF and linear kernel. These findings agree with the literature, that reported the activation response to neutral stimuli in different brain areas to be associated with MDD [88, 137]. Unsurprisingly, we also note that the beta04 predictor, corresponding to the break condition, achieves the worst result. Furthermore, we observe that using an average of the *beta* image is comparable to the best performing (rbfSVM) or inferior to it (linSVM.; Technically, imaging average can be seen as a multi-modal analysis, nevertheless, no specific multi-modal approach was yet used. Overall, we do not notice a clear difference among the different choices of beta images, while the predictive performance is superior to a random classifier in every condition. As a next step, we compare the SVM with other linear and non-linear classifiers: k-NN, LDA, RF and LR. Given the similar performance of linear and RBF kernel, and considering the crucial importance of interpretability, we pursue our analysis with respect to the linSVM method only. In this experiment, we further apply the *F-score* feature selection method. Results on the beta02 image are shown in Figure 3.8. We are positively surprised to observe that two of the linear methods, linSVM and LR show the highest predictive power. Overall, we obtain the worse performance with *k-NN*, probably due to the Euclidean distance measure being inappropriate as a similarity score in the images. Indeed, despite the spatial imaging structure is currently being ignored, it would certainly be interesting to include this component into our analysis, and a distance based classifier as the *k-NN*

Figure 3.8: Classifiers comparison on the neutral stimuli condition. Performance on the neutral stimuli condition with different classifiers and f-Score feature selection. The left plot shows accuracy precision and recall for each classifier. On the right side, the ROC curve is reported.



will benefit the most from it. Despite *beta* images providing predictive and meaningful results, it would be more difficult to extrapolate a satisfying clinical interpretation from the weight maps. Nonetheless, *contrast* images produce a straightforward understanding of the obtained patterns with respect to the features importance, as the voxels reflect the difference in reaction between experimental conditions across groups of individuals. We are especially interested in contrasts defining the activation difference between a stimuli and break condition. We compare th results of three feature selection based variants of the linear SVM model: SVM-fScore, SVM-RFE, and SVM without feature selection (SVM-wFs). Classification performance is reported in Table 3.4. We observe that the MVPA method, and in particular SVM-fScore, yields significant classification accuracies in all the working memory condition versus break contrasts. With a 73.74% accuracy, the *Neu* > *Break* contrast reports the best results, as it is further confirmed by the highest sensitivity and specificity. The *Neg* > *Break* also gives good predictive performance, with accuracy sensitivity and specificity all above 70%. Slightly lower but still significant results are identified when the whole WM conditions or the positive condition against the break are considered as input image. The classifiers trained on the *Emo* > *Neu* contrast, that only evaluates differences between conditions (and no break), is comparable to a random one, yielding non-significant prediction for each of the three variants of our pipeline. Overall, we observe that the SVM-fScore method is the most successful for analysing this cohort, accordingly we restrict our post-hoc analysis to it. Additional results with 10 fold CV, for comparison and validation purposes, are reported in Appendix A.

Table 3.4: Classification results on the contrast of interest with linear SVM and different feature selection strategies. An asterisk * depict significant classification result based on the permutation test ($p - value < 0.001$).

	SVM-fScore			*SVM-RFE*			*SVM-wFs*		
Contrast	Acc	Sens	Spec	Acc	Sens	Spec	Acc	Sens	Spec
WM > Break	66.10*	68.42	63.93	64.41	68.42	60.66	62.72	66.66	59.02
Pos > Break	63.56*	68.42	59.02	61.86	70.18	54.10	60.16	66.66	54.10
Neg > Break	71.18*	71.93	70.49	62.25	70.18	60.66	64.41	68.42	60.66
Neu > Break	73.73*	71.93	75.41	72.88	80.70	65.57	66.94	70.17	63.94
Emo > Neu	49.15	64.91	34.43	50.00	75.44	26.23	39.83	57.89	22.95

SITE EFFECT. The classification accuracies on the three scenarios introduced to asses the influence of the acquisition site, suggest that similar performance is obtained when using samples from either of the two sites. In scenario (i), where the evaluation step is performed on each site separately, we obtain an accuracy of 64.29% and 66.67% on Zurich and Berlin, respectively. This suggests that our method gives similar performance when we evaluate on each of the two sub-samples, showing that it is not biased towards one of the two groups and the learning power of the model is homogeneous across images from different domains. In the second case (ii), where training and evaluation are both treated independently on each sample, the accuracy is 57.14% on Zurich and 68.89% on Berlin. Remarkably, this indicates that, as expected, a lower sample size (Zurich) is negatively affecting the ability of the model to perform accurate predictions. Lastly, when training is performed on Berlin (larger sample size) and evaluation on Zurich, the accuracy is 60.71%, demonstrating that our classifier trained on a different (larger) sample (Berlin) can successfully be adapted for out–of–sample predictions on a smaller domain (Zurich). We conclude that the site does not particularly affect the analysis, while the specific sample size plays a crucial role in enhancing the performance. It must be remembered that these results are only valid in the context of a post-processing analysis and should not be interpreted as informative of the sample generalisation performance.

POST-HOC ROI We conclude the MDD study by evaluating the post-hoc ROI cluster analysis based on the weight map obtained by SVM-fScore on the *WM>break* contrast. Despite not providing the highest classification accuracies, the *WM>break* contrast is the most relevant in a psychiatric evaluation. In fact, previous studies confirmed the existing differences in BOLD activation values between depressed and healthy subjects, when performing a working memory task. More importantly, the literature reported inconsistent results regarding the activated area and regions, while we aim to overcome some of these limitations by considering a larger sample size, using a combination of the WM tasks (*WM>break*) and employing an MVPC pipeline. The cluster analysis resulted in 14 regions, half of which depicted higher activity in the DP vs HC, and viceversa. The average contribution of each region with respect to the SVM weights is

Figure 3.9: Average SVM weights in ROIs. The average value of the weights ($\times 10^{-3}$) in each region as obtained by the SVM-fScore classification is shown. The dotted lines show the average of all positive and negative weights in the whole brain.

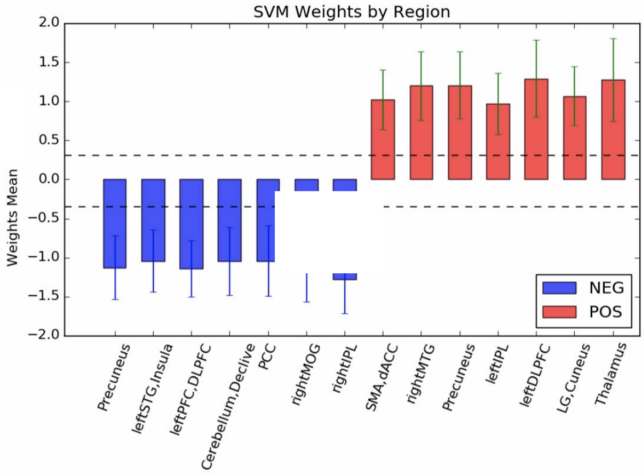
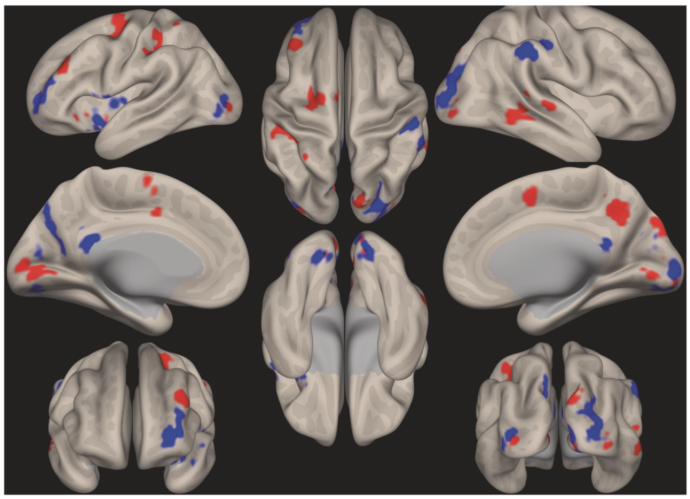


Figure 3.10: SVM weight map. The location of the most relevant SVM classification weights from the WM > fixation contrast are shown (20% of the highest weights with a cluster threshold of 50 voxels). Red regions depict more activation in MDD patients. Blue regions depict more activation in healthy controls.



reported in Figure 3.9 and their correspondence in the brain is shown in Figure 3.10. The post-hoc results suggest that the majority of regions with discriminative power were located in the default mode network (DMN) and dorsolateral prefrontal cortex (DLPFC) brain area, which are typically involved in cognitive control. We refer the interested reader to Gärtner *et al.* [63] for an extended discussion on the clinical findings.

MULTI-MODAL IMAGING ANALYSIS. Finally, we analyse the MDD cohort from a multi-modal perspective. We evaluate the performance of EasyMKL and avgMKL with linear and RBF kernels, as well as the feature concatenation baseline. Results are reported in Table 3.5. Because of the exploding cardinality when considering all the possible

Table 3.5: Classification results of the multi-modal analysis.

Modality	linFC	rbfFC	linAVG	rbfAVG	linEASY	rbfEASY
Con04	69.49%	**71.19**%	68.64%	**70.34**%	68.64%	70.34%
RS	66.10%	69.49%	68.64%	64.41%	68.64%	64.41%
Struct	63.56%	50.85%	65.25%	50.85%	65.25%	50.85%
Con04 + RS	69.49%	68.64%	68.64%	68.64%	67.80%	**72.88**%
Struct + RS	65.25%	55.08%	68.64%	60.17%	**70.34**%	62.71%
Con04 + Struct	63.56%	61.02%	64.41%	55.93%	66.10%	57.63%
ConAll	**72.88**%	61.02%	**70.34**%	64.41%	**70.34**%	67.80%
ConAll + Struct + RS	64.41%	61.02%	61.86%	61.02%	64.41%	56.78%
Con04 + Struct + RS	68.64%	61.02%	66.94%	63.56%	68.64%	58.57%

modalities combination, we restrict to an informative subset by aggregating the best performing (*Neu* > *Break*) contrast with resting state and structural data, as well as with a combination of all the contrast images. It can be observed that a combination of all the contrast images performs the best in all the linear methods. On the other hand, with the RBF kernel, the optimal contrast image already achieves the best accuracy without additional data integration. With EasyMKL and RBF kernel the resting state image with the *Neu* > *WM* contrast gives the higher result, with a 2% improvement over *con*04 and an 8% improvement over RS. Nevertheless, we overall observe that there is no clear benefit in using more complex MKL methods over simple feature concatenation strategies, and sometimes even as compared to the uni-modal setting.

MULTIPLE SCLEROSIS STUDY

For the MS study we use the ROI based extracted features to compare the classification performance of the MS vs HC task with diffusion and MTR metrics. The comparison includes a linSVM, rbfSVM and RF classifiers. For the multi-modal approaches we compare each of these models to feature concatenation and to EasyMKL, with RBF and linear kernels. Figures 3.11 and 3.13 summarise our findings by showing the precision recall curves, and in particular evaluating the AUPRC. The choice of this particular evaluation is motivated by the highly unbalanced sample in our cohort (see Section 3.1.2). We clearly observe that either with single modalities or via concatenation of them, the linearSVM performs quite poorly. However, the non–linear approaches including the rbfSVM and RF provide a clear improvement in the predictive performance in both the uni–modal and multi–modal scenario. Nevertheless, the feature concatenation ap-

proach does not seem to be beneficial in this case, leading to either no improvement with respect to the best performing modality or to a performance decrease. Looking at the EasyMKL methods (Figure 3.13), again we see that the linear kernel has a worse performance than the RBF. However, in the linear case, we note a clear benefit of using MKL as compared to the naive feature concatenation or uni-modal setting. For the RBF this is not as clear, although it is evident that the worse performing modality MTR, does not affect the prediction and the results are either dominated by the best modality or outperform the isolated contribution of individual images.

3.5 Discussion

Study limitations. We are certainly limited by several factors affecting the results and interpretation of our findings, either inherent within the MRI data or specific to our cohort. The low sample size is a huge limitation on the learning power of the machine learning models. We partially overcome this issue by employing a LOOCV strategy. Besides, classical machine learning algorithms such as SVM or RF are known to still perform well in low data availability regimes. Furthermore, the phenotype assignment is a problem both in MDD and MS, and the uncertainty of the diagnosis is also reflected on the assessment of disease severity and subtypes. These issues are particularly problematic for border line cases, where a binary label assignment could be inadequate.

Uni-modal versus multi-modal. A remaining open question is whether adding more modalities is beneficial for the classification task. So far, our results show conflicting outcomes, but overall we conclude that in this cohort there is no clear improvement in the performance when multiple MRI modalities are combined. Certainly, our findings have to be seen in the context of the current data and they are particularly limited by the small sample size. Indeed, the more complex the model and the more features are included, the larger is the negative impact of limited data availability. Actions can be taken to overcome these limitations, for instance employing transfer learning strategies would be a natural follow up, as we argue in Chapter 7.

Figure 3.11: Precision-Recall curves. Comparison of uni-modal and multi-modal classifiers. Feature concatenation is used for the multi-modal approaches

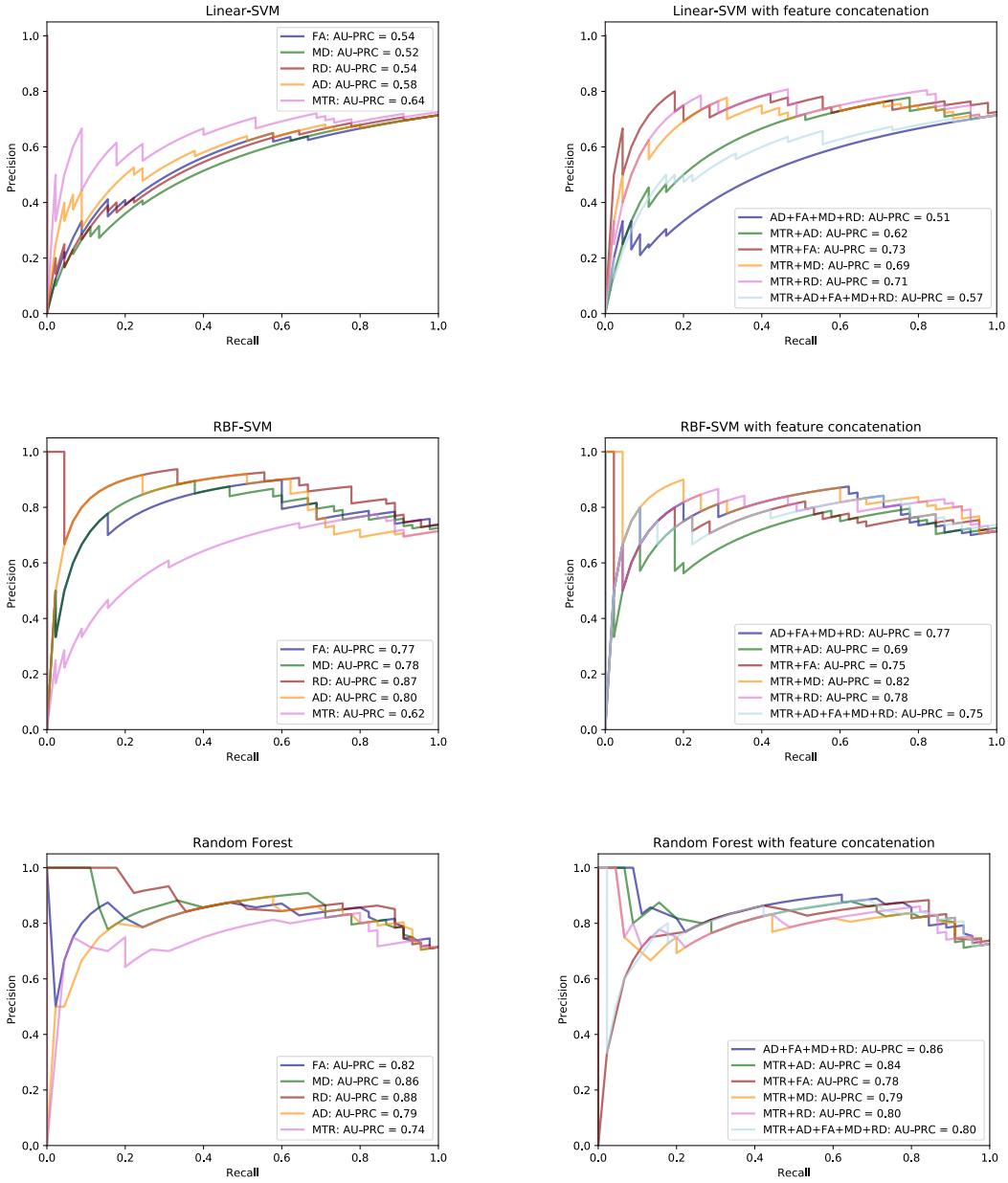
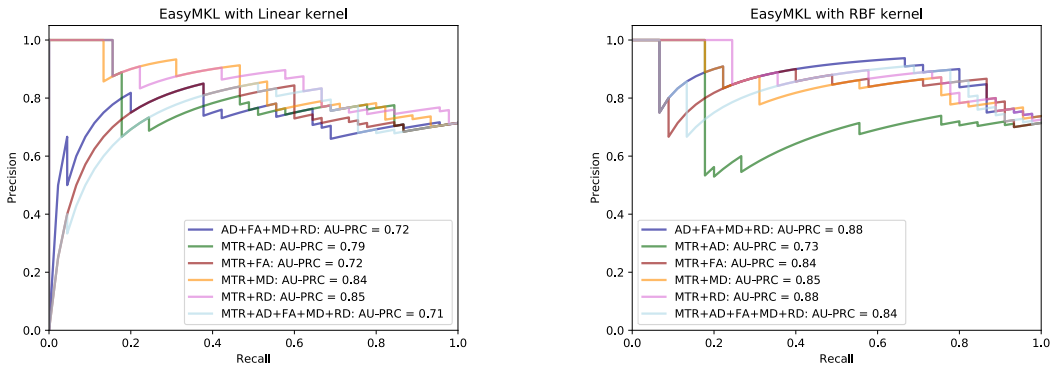


Figure 3.13: Precision-Recall curves. Easy Multiple Kernel Learning with Linear and RBF kernel.



4 Predicting complex tasks

The previous chapter mostly focused on patient control phenotype prediction, and on understanding the effect on the classification performance of multiple MRI modalities integration. Nevertheless, in clinical studies the patient control task is typically among the most trivial for a predictor. Especially for neurological disorders, the assessment performed by a clinician who evaluates visible symptoms and behavioural hints occurs at a stage where the health status is already clear. For example, in MS the brain alterations are visible by eye on the MRI, so the benefit of an automated algorithm are minimal. In contrast, more complex prediction tasks such as disease progression, early diagnosis, or treatment response are harder to assess in the clinic. In general, the symptomatic evolution of the patient is not easy to forecast at an early stage. In recent years, many efforts towards personalized and precision medicine have been taken within the machine learning community, aiming to support the physicians in early intervention and personalised treatments. The recommendation automated systems exploit the complex feature interactions to predict the task. For instance, one could be interested to learn the best drug treatment to apply on a single patient to maximise the probability of a positive response. Thus, the algorithm would perform the prediction based on a combination of clinical features and historical data, by discovering hidden connections that might be missed by the human expert. Unsurprisingly, acquisition and analysis of MRI data have played a crucial role in this development, representing an additional source of information for the doctors and a valuable input for the automated systems.

In this chapter, we extend our investigation of the MDD and MS studies to complex neurological tasks. Specifically, we examine the response to Electroconvulsive Therapy (ECT), an effective and yet aggressive procedure to treat certain psychiatric conditions, on a subset of the MDD cohort. For the MS study, we explore the capacity of unsupervised learning algorithms to cluster the MS patients into disease subtypes and discuss the task of predicting the course of MS.

The reminder of this chapter is organised as follows. In Section 4.1 we examine the MDD treatment response prediction task. We introduce the data and problem, discussing the difference between a regression and classification based approach, and subsequently present our experimental findings. Section 4.2 describes the MS study and results, presenting an unsupervised based approach to tackle the disease subtype categorisation. Our discussion in Section 4.3 addresses the limitations of the current study and explores directions for further investigation.

4.1 Treatment response prediction in depression

4.1.1 Data and feature extraction

We begin by describing the data type and feature extraction steps. Since the MDD cohort partially overlaps with the one introduced in Chapter 3, we will omit a complete description and solely focus on the additional details relevant for this context. A more comprehensive overview can also be found in [65].

We consider a set of patients diagnosed with Major Depressive Disorder (MDD; see Section 3.1.1), our goal is to assess their response to Electroconvulsive therapy (ECT) using MRI imaging data as input. This is a retrospective study, implying that the response has been already recorded, while the MRI scan has been performed before ECT. This image prior to treatment will be used for the prediction task. While ECT is a very successful therapy for severe depression cases, with a response rate between 60%-80%, it is still a very demanding procedure and can bring several side effects, including memory loss [11, 29]. Motivated by this rich and interesting nature of the treatment, we deem as a very important task to be able to provide accurate recommendations to the psychiatrist on how the subject will respond to the ECT. To solve this problem, many studies have been based on demographics and clinical factors, such as psychotic symptoms or depression severity [177, 187]. Other work exploited biological information, in particular MRI based biomarkers [44, 143]. Recently, an approach using structural MRI and Grey Matter Volume (GMV) features in a machine learning framework achieved very high classification performance [149]. Inspired by this work, we first aim to replicate this study in our larger cohort. Subsequently, we extend the investigation to predict a continuous percentage change value in the disease severity, therefore turning the classification problem into a regression task. Indeed, defining a binary label as responders and non-responders is often problematic since several patients only showed a partial response.

Electroconvulsive therapy. ECT was conducted 3 times a week with a total of 12 sessions. Patients showing a partial response received further ECT sessions, until they did not show improvement any more. Physiological monitoring included two–lead electroencephalogram (EEG), electromyography (EMG), electrocardiogram (ECG), blood pressure and oxygen saturation. Initially, the stimulus intensity to set the seizure threshold was 5%, then treatment was performed with an intensity 7 times higher than the threshold. If the seizure activity was less than 20 seconds in EEG, in the following ECT sessions the stimulus intensity was raised in steps of 5-10%.

Clinical assessment. Depression severity was assessed weekly during the treatment with the Montgomery-Asberg Depression Rating Scale (MADRS; [125]). Furthermore, to evaluate symptom severity the Beck Depression Inventory second version (BDI-II; [13]) as a self-report was used before and after ECT. To quantify the treatment efficiency, the percent of symptom reduction (PSR) was calculated as percent change from MADRS baseline ([MADRS score after ECT – MADRS baseline score] / MADRS

baseline score x 100), and patients with a PSR greater or equal than 50% were classified as responders. Hospital discharge summaries where also checked for the final assessment, to verify that the improvement was not due to other treatments and to verify possible inconsistencies.

Data selection

The current study consists of 71 patients, including 41 females (age: 50.72 ± 17.66; age range: 19-90) and the remaining matching males, diagnosed with a major effective disorder and that received an ECT treatment. ICD-10 codes were used as diagnostic criteria to select the subjects of interest, which were recruited at the Department of Psychiatry, Charitè Universitätsmedizin Berlin, Campus Benjamin Franklin, Berlin (CHAR) between 2012 and 2018. In particular, 66% of the subjects are classified with severe recurrent major depressive disorder (F33), but other related disorders are also included (F32-14%, F31-13%, F25-4%, F34-3%). Patients who received a structural MRI before ECT treatment and simultaneously receiving antidepressant medication were included. Exclusion criteria were based on incomplete data only. The study was approved by the institutional review board at CHAR.

Data acquisition and preprocessing

A structural MRI (sMRI) scan was acquired for all subjects before ECT, as part of a clinical routine. The scan was performed on either a 1.5 Tesla scanner (Magnetom Aera, Siemens Healthineers, Erlangen) or a 3 Tesla scanner (Magnetom Skyra, Siemens Healthineers, Erlangen) both equipped with a 20 -channel head/neck surface coil. All patients were scanned head-first in supine position using a 3 D isotropic high-spatial resolution T1-weighted Turbo-Flash sequence with near identical sequence. Parameters were set as follows: TR: 1900 - 2200 ms; TE: 2.49 - 2.88 ms; TI: 900 ms; Flip-angle: $\leq 15^{\circ}$; voxel size: $0.9 \times 0.9 \times 0.9$ mm^3 at 3 Tesla or $1 \times 1 \times 1$ mm^3 at 1.5 Tesla; number of excitations: 1; parallel imaging with an acceleration factor of 2 (GRAPPA algorithm [76]); acquisition time: 4.23 - 4.56 minutes. All the images were preprocessed with the CAT12 [1] toolbox [67] implemented in SPM 12, as described in Section 3.1.1 and Appendix A. The result of the preprocessing steps is a gray matter volume (GMV) probability map of the brain per subject, that allows to quantify the presence of GM tissue in the brain.

4.1.2 Methods

We investigate the ECT response prediction task from the two perspectives, either as classification or regression. In the classification task, we used the binary label as assigned via clinical assessment. For the regression problem, we used the PSR as a percentage of symptom reduction, where 0% implies no improvement after ECT and 100% means full recovery. In practice, we observed that most of the response rates lie in the middle range and are centred around 50%, making the classification task particularly

[1] http://www.neuro.uni-jena.de/cat/

problematic due to the uncertainty of the assigned labels. A regression based analysis can soften this issue by predicting a continuous response, and possibly providing the clinician with a more informative measure.

We first briefly review the multivariate pattern analysis (MVPA) approach introduced in Section 3.2.2, and subsequently extend it to the regression scenario.

MVPA pipelines extend machine learning models to MRI imaging analysis, with the major objective to extract MRI based features or interactions of them, that are jointly informative for the clinical interpretation and predictive for the learning task. The main advantage of a multivariate approach over the univariate counterpart, is indeed to allow for dependency between features (Section 3.2.1). Either the entire image or a subpart of it, determined by a feature selection strategy, represent a valid input for the prediction model. The MVPA will outcome the optimal prediction and a set of features that are mostly informative with respect to group separation (classification) or associated to a continuous prediction target (regression). These features translate into patterns of activity, eventually located in distant brain areas. In this study, the classification problem is to distinguish between responder and non-responders, while the regression task corresponds to estimate the reduction of symptom severity. In general, MVPA methods should be equipped with cross validation techniques, to guarantee generalisation ability on an independent sample with similar characteristics [119]. In the next sections, we will present in details the machine learning pipeline on the ECT prediction task. An overview is shown in Figure 4.1.

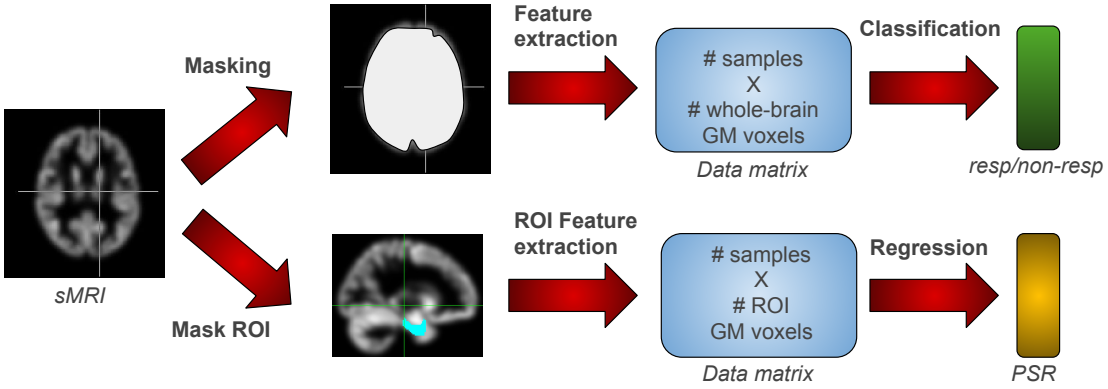
Feature extraction

The first step of both the classification and regression pipeline is to extract MRI features that are informative for the task of interest and representative of the sample cohort. For this study, we use probability maps of the GM, that is the voxel values obtained from the sMRI. We will also refer to these features as GMV, since the voxel values ultimately return the grey matter volume when aggregated. To guarantee that the extracted voxels are grey matter we compute a brain mask, where all the values below 0.1 are considered as non-GM tissue. To obtain the group mask we intersect single subject masks. Both whole brain and regions of interest are considered, leading to the following two types of features:

(i) Whole - brain GMV: the GM map of the whole brain tissue, resulting in an input matrix of size $X_{whole} \in \mathbb{R}^{71 \times 469386}$, after masking and GM voxel filtering;

(ii) aPHCr GMV: the GM map of the anterior right parahippocampal gyrus (aPHCr) region, extracted with the FSL Harvard-Oxford Atlas and resulting in a data matrix of size $X_{aPHCr} \in \mathbb{R}^{71 \times 1679}$.

Figure 4.1: Machine learning pipeline for the classification and regression analysis. Input for all predictive analyses conducted is the smoothed modulated whole-brain gray matter volume (GMV). Steps in the classification analysis: Masking, using a whole-brain coverage mask. Feature extraction using SVM-fScore. SVM-based classification using responders (resp) vs non-responders (non-resp) class labels. Steps in the post-hoc regression analysis: Masking, using an anatomical ROI mask of the right anterior parahippocampal gyrus (aPHCr). Feature extraction corresponding to GM voxels in the ROI mask. Linear regression to predict the percentage of symptom reduction (PSR).



Classification

For the classification task of predicting responders versus non-responders, we employ the *SVM-fScore* approach as described in Section 3.2.2, using as input the whole - brain GMV features.

Post-processing cluster analysis. As discussed in Chapter 3, we are particularly interested to obtain classification results that are of relevance to the clinician. We used the weight maps extracted from the linear SVM in order to identify brain regions that are mostly contributing to the classification task. To this end, we retain 5% of the highest absolute weights and apply a cluster threshold of 500 voxels, to focus on anatomically meaningful clusters. As observed in the previous chapter, it is meaningless in an MRI image to provide an interpretation of a single isolated voxel activation, but we are interested in finding brain areas that characterize the task of interest. The use of linear separation boundaries in the SVM allowed for a straightforward interpretation of the feature weights, implying that higher absolute values corresponded to the most discriminative features. Due to the applied label convention in the classification model, a positive weight sign indicates higher GMV in responders, and a negative weight sign indicated higher GMV in non-responders.

Regression

It is natural to extend the MVPC methods (Section 3.2.2) to a regression model, such as an Support Vector Regression or a simple multivariate linear regression. In our work, we use multivariate linear regression analysis to predict the PSR with respect to the MADRS score, that is:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_1 x_{i,2} + \ldots + \beta_p x_{i,p} + \epsilon_i, \tag{4.1}$$

for each $i = 1, \ldots, n$. Here, n is the number of samples, p is the number of features (e.g. voxels), β are the parameters, y is the response variable (e.g. PSR) and x is the explanatory variable, for instance the vector of voxel values. We assume that the noise term ϵ is normally distributed with constant variance and uncorrelated, i.e. $\epsilon \sim \mathcal{N}(0, \sigma^2)$. As for the classification pipeline, we can perform a feature selection step by using some regression based criteria. We replace the *F-score* in classification with and *F-test* for regression. First, the correlation coefficient between the response and feature is computed, as:

$$\rho_i = \frac{(x_{\cdot,j} - \bar{x}_{\cdot,j}) \cdot (y - \bar{y})}{(s_j, s_y)} \tag{4.2}$$

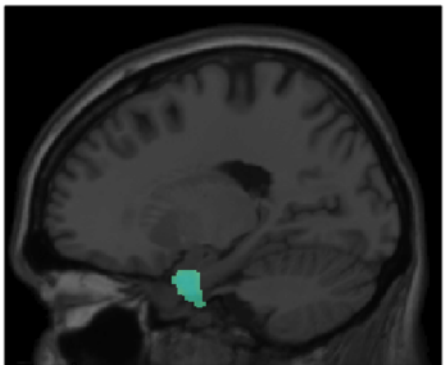
where s_j and s_y are the sample standard deviation of feature j and y, respectively. Then, we can use an *F-test* to obtain a *p-value* and rank the feature importance accordingly. In the regression study, we used GMV aPHCr features as input.

4.1.3 Experiments

Experimental setup

Classification. In the classification analysis the whole–brain GMV features were used as input for the machine learning pipeline. A leave–one–out cross validation (LOOCV) was employed to evaluate the classifier performance, as recommended for limited sample size domains [179]. The tuned hyperparameters, i.e. the C in SVM and and the percentage of selected features, are learned on the training data only via 5 fold cross validation. For the SVM parameter, the grid $C = \{10^{-5}, 10^{-2}, \ldots, 10^{1}\}$ was used, while to select the number of features, varying percentages in the range $\{10\%, 20\%, \ldots, 50\%\}$ were evaluated. The classification weight maps for subsequent analyses were constructed by averaging the weights over all folds of the cross validation. The performance of the classifier was evaluated in terms of accuracy, sensitivity, and specificity, and the statistical significance of the classification accuracy was assessed via permutation test, with 1000 repetitions (see Section 3.4.1 for details on these procedures).

Figure 4.2: Classification weight maps. Blue colour depicts a cluster of the most contributing classification weights in the right anterior parahippocampal gyrus (aPHCr; MNI coordinates of center: 18, 0, −30; size: 573 voxels).



REGRESSION. In the regression analysis the aPHCr features were used as input. The predictive performance was evaluated with a LOOCV approach, minimising the mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.3}$$

the features were selected on the training data only via inner 5 fold cross validation, in a range of $\{10\%, 20\%, \ldots, 100\%\}$. To evaluate the performance, we use the Pearson correlation coefficient (PCC) between true (y) and predicted ($\hat{y}$) PSR:

$$\rho(y, \hat{y}) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}. \tag{4.4}$$

RESULTS

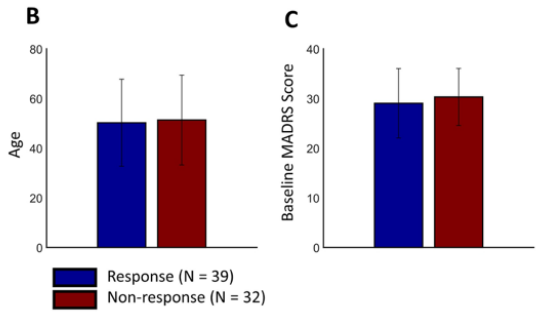
CLASSIFICATION. The SVM-fScore approach was applied on the 71 patients, of which 39 positively responded to ECT and 32 are non-responders. We obtained a classification accuracy of 69.01%, with a sensitivity of 66.67% and specificity of 71.87%, that is 26/39 responders and 23/32 non-responders were correctly classified by SVM-fScore. The permutation test ($p\text{-}value = 0.008$) revealed that structural MR images in our sample provided enough signal and information to distinguish between responders and non-responders. Subsequently, we performed the post-processing cluster analysis, which showed that a GMV cluster in the right anterior parahippocampal gyrus (aPHCr) provided most informative contribution in the characterization of ECT response (Figure 4.2).

We also evaluate the performance obtained by simple clinical predictors, which are known to be related to the response task, and use these as input features for the SVM-fScore, either in a univariate or in a multivariate fashion. Outperforming the clinical feature baseline is crucial to affirm the importance of acquiring good quality MRI images. Results showed that no statistically significant predictive power was observed

Figure 4.3: Descriptive statistics of clinical predictors in ECT responders and non-responders. A: Presence of psychotic symptoms. B: Age at start of ECT. C: Baseline depression severity (MADRS score).

A

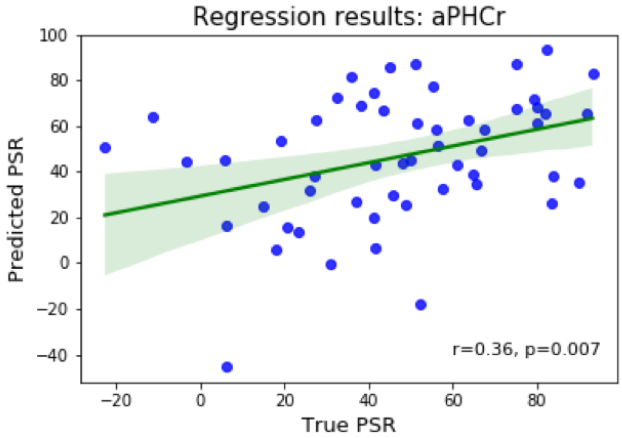
	Psychotic symptoms		
	Yes	No	Total
Response	9	30	39
Non-Response	6	26	32
Total	15	57	72



with respect to *age*, *presence of psychotic symptoms* and *depression severity*. We also observed that in most of the CV splits, and for almost every sample, the classifier was predicting the majority class (responders). This suggests that no significant information is hidden in the clinical variables to provide a meaningful separation. Classical statistical analysis supports the machine learning findings, given that no differences between ECT responders and non-responders is observed for psychotic symptoms (chi-square, $p = 0.61$), age (t-statistic, $p = 0.72$), and depression severity (t-statistic, $p = 0.65$), as shown by the descriptive statistics in Figure 4.3.

REGRESSION. To predict PSR changes, we used a subset of the original data consisting of 54 patients, selected by availability and quality control. We used an anatomical mask of the aPHCr and apply a multivariate regression analysis on the GMV voxel values from this region. The aPHCr has been previously associated with depression and also showed discriminative power in our post-hoc classification analysis [202, 203]. Results are presented in Figure 4.4. In the figure, the green line represents a linear regression fit of the predicted outcome against the true PSR, with the shaded green area highlighting a confidence interval for the regression slope parameter. The findings show a positive significant correlation ($r = 0.36$; *p-value* $= 0.007$) between the predicted PSR on aPHCr and the corresponding true percentage variation. Therefore, we conclude that a signal in the GMV values of the aPHCr region can be identified as significantly associated with the PSR changes after ECT.

Figure 4.4: Regression results on aPHCr region. Predicted PSR versus true PSR with a linear regression fit. Green area is a confidence interval for the slope parameter.



4.2 Identifying patient subtypes in Multiple Sclerosis

For the MS study, we used the same data cohort and modalities as described in Section 3.1.2, with analogous preprocessing and acquisition steps. We investigate the complex tasks of identifying patient subtypes and establish the progression of MS. Both scenarios are very challenging from a clinical perspective, given that the phenotype exhibits high uncertainty. For the MS subtype, we recall our distinction across RRMS, SPMS, PPMS. Despite the subgroups being clinically definable, the symptoms and evolution of MS might vary across subjects, and a patient may be assigned to a different group at a later stage of the disease, leading to the uncertainty of sub-group assignment. This uncertainty, certainly reflects in the disease progresion task as well. Here, the challenges come from the unstable disease evolution, on one side, and from the clinical assessment evaluation, on the other side. We consider the MS subtype task in an unsupervised fashion. Namely, we aim at clustering patients based on the MRI scans, to ultimately assess if the established separation is informative of the MS type. Of course, we aim for the patients belonging to the same MS type to be assigned to the same cluster.

Data and feature extraction. The MS cohort, including the acquisition and preprocessing protocol was established in Section 3.1.2. As input for the analysis, we also use the same ROI based features extracted from the diffusion metrics and MTR images, as detailed in Section 3.3.2.

4.2.1 Methods and results

To analyse the power of unsupervised algorithms for the MS subtypes task, we perform a comparison of different clustering approaches. The number of subgroups to

consider, corresponding to the number of clusters, is particularly tricky to assess. In principle, it should be treated as an unknown parameter and selected using established evaluation criteria, for example a measure of consistency or cluster similarity. Several techniques exist for assessing the optimal number of clusters. For instance, one can use cross–validation based methods, that evaluate the performance of cluster similarity on test data. Another possibility is the *Silhouette coefficient*, which employs the distance between and within cluster objects to evaluate the quality of the separation. The *Elbow method* is also a valid alternative, as it looks at the explained variation of the data with varying number of clusters. To facilitate the clustering task, already challenging from a clinical and analytical perspective, we omit this step and rather perform a retrospective investigation. In practice, we assume to know how many subgroups (clusters) are defined in our cohort and we wish to understand how well the model can retrieve those.

CLUSTERING. We use traditional clustering methods including Spectral Clustering [186], DBSCAN [48], K-Means and Hierarchical Clustering [54]. We also conduct a principal components analysis (PCA [54]) as a preprocessing step on the ROI features and apply clustering on the reduced data, to facilitate interpretation and visualisation and potentially peak patterns from the PCA transformed features that might be hidden in the original space.

EVALUATION. For clustering evaluation, we use the Rand Index, and in particular the adjusted version, which is a classical measure of similarity to evaluate the overlap between the true and assigned cluster label. The Rand index is indeed equivalent to accuracy, and can be defined in terms correct and wrong prediction counts as:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}. \tag{4.5}$$

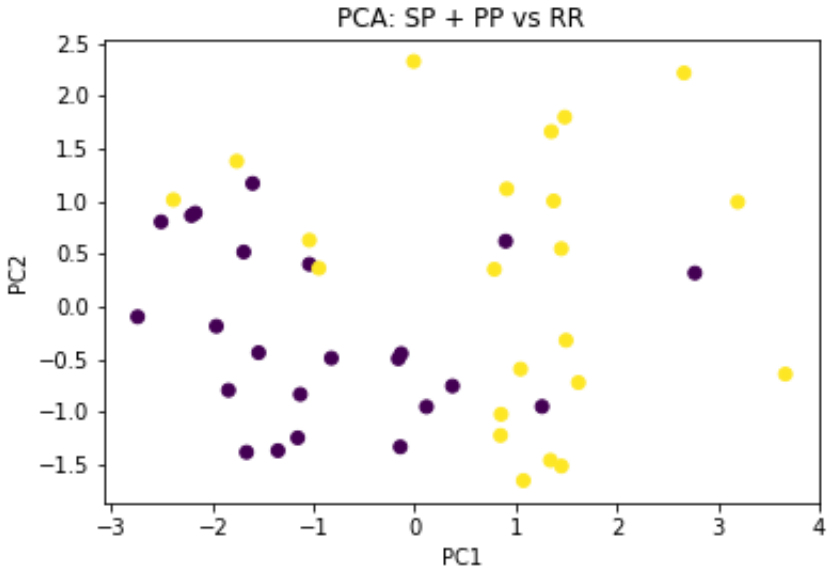
The adjusted-rand Index is a corrected version of RI. The adjustment is made with respect to the expected similarity of the comparison, in pracitce correcting for random chance. This is defined as:

$$ARI = \frac{RI - Exp(RI)}{(max(RI) - Exp(RI))}. \tag{4.6}$$

where $Exp(RI)$ is the expected outcome of a random algorithm.

RESULTS. Overall, clustering into the 3 MS sub-types did not provide satisfactory results. When comparing across different input metrics, we observed that the best overall results are achieved with the FA images, reporting the following accuracies: k-means ARI= 0.23; hierarchical ARI= 0.28; spectral ARI= 0.20. The DB scan was unable to pick any meaningful signal, classifying all the points as "noisy". This minimal performance suggested that the ROI features are unsuitable to provide any distinction among

Figure 4.5: First and second PCA of the ROI features. For the SP+PP group (purple) versus RR (yellow).



the MS types. Therefore, in our subsequent step we aim at reducing the complexity of the model by considering a binary separation. In particular, given that the PP patients are a minority (5%) and they belong to the set of the most severely diagnosed subjects, we merge them with the SP, generating the new separation task $SP + PP$ vs RR. With this approach, we clearly observe that more signal can be retrieved. While the spectral and DBSCAN clustering still outcome non significant clusters, with k-means and hierarchical clustering we achieve an accuracy of 65% and 74%, respectively. Lastly, we evaluate the effect of performing PCA and clustering on the first two principal components only. This analysis also reveals more promising results. In figure 4.5 we show the first and second principal components and label the points according to the MS. It is evident that a good separation of the two groups is achievable. Running a hierarchical clustering analysis, we also observe a pretty good recovery of 77% between the true and assigned MS type.

4.3 Discussion

4.3.1 Treatment response

Our classification results confirm that there is predictive power in structural brain images of ECT patients. This is further supported from the analysis of the weight maps, which showed that a region in the right anterior hippocampal cortex contributed most

to the prediction. These findings align with the literature, as GMV increases in the hippocampus after ECT, while patients with less GMV in this region are more likely to respond to the treatment [154]. We also showed that, compared to the clinical predictors lacking statistical signal (Figure 4.3), MRI biomarkers were successfully able to predict the response to ECT. These considerations raise the question of whether it would be beneficial to integrate additional MRI modalities to further improve the prediction performance. Our findings in the previous chapter would suggest that redundant signal could indeed negatively affect the model learning capabilities. Nevertheless, those results referred to a different and simpler prediction task (MDD vs HC), while in this complex scenario it is plausible that hidden interactions can be extracted via combination of multiple MRI features. Lack of data in our cohort did not allow to explore this hypothesis, but we are confident to recommend that future studies should consider MKL and multi-modal integration for the ECT task.

Another major discussion point concerns the choice of the dichotomisation of the response label, turning the PSR into a binary prediction task, motivating us to explore the regression perspective. Indeed, as we can observe from Figure 4.4 the majority of either true and predicted value are centred around 50%. This poses an issue to the binary response phenotype, since with this threshold many subjects lie at the boarder of the two classes. While our results are yet not optimal, given the large prediction interval, we believe that future work should investigate the regression based approach in more details. Obtaining narrow intervals and accurate PSR predictions is certainly among the most relevant information to provide to a clinician. The choice of the modality and small sample size are possible reasons to justify our modest results. Nevertheless, our correlation coefficient appears to be close to the 0.05 significance level, a promising outcome that encourages further investigation.

4.3.2 Complex tasks in Multiple Sclerosis

The classification of MS subtypes is particularly challenging, as well as it is often unclear for a medical expert to assign the patient to the correct group, mostly due to the unpredictable course of MS. We performed a clustering based analysis to evaluate the prediction power of diffusion MRI metrics in the patient subgroup separation task. Overall, our results confirm the difficulty of determining a correct label assignment to a specific subgroup. We verified that only a subset of the ROI features was associated with the MS subtype phenotype, suggesting that using the whole ROI data might be not necessary, if yet uninformative. This speculation is confirmed by the PCA analysis, where we observed that in a low dimensional space it is easier to identify clusters of patient subgroups. However, our study should be interpreted in retrospect since the number of clusters was considered to be fixed. Future work should focus on optimising this parameter given that, a priori, the distribution of the MS patients across the subtypes is unknown. Of course, a larger sample size is required for this investigation, as opposed to our cohort where each group only could use an handful of subjects, resulting in the learning capabilities of the algorithm to be limited.

An additional complex and crucial task in MS is the assessment of disease progression. There are several ways to evaluate progression, for example looking at established criteria as the EDSS score or at the volume of the white matter lesions in the brain. Both are assessed by experts, either via clinical investigation or via observation by eye on the image itself. While we did not report these results, the preliminary classification and statistical analysis within our cohort suggested that very little signal could be retrieved from the available data to successfully solve the progression task. We certainly envision that future work could benefit from an extended data acquisition and we recommend to extend the uni– and multi–modal approaches developed in Chapter 3 to the MS progression task.

Part III

Learning on graphs

5 Wasserstein Weisfeiler-Lehman graph kernels

In previous chapters, we focused on applications of kernel based methods on MRI imaging analysis. We showed how kernels can be integrated in classical machine learning algorithms and combined to optimally learn information from different data modalities. We will now shift our attention to kernel applications on the graph structured data domain, as we presented in Chapter 2. Most of the these graph kernels rely on the $\mathcal{R}$ – convolution framework [82], based on a decomposition of structured objects into substructures defining local similarities, which are then aggregated to compute the final similarity score. However, one of the major limitations of these approaches lies in the naive aggregation step, which is generally a sum or average. For example, in the popular WL kernel one–dimensional node features are summed to obtain a graph representation and ultimately the kernel value (Section 2.2). Therefore, complex structural similarity and non–linear dependencies across node representations are ignored at the graph level, resulting in a potential loss of information. Indeed, the simplicity of the *readout* step (see Section 2.3) is also an issue in many graph neural network approaches, where the complexity of the model needs to be controlled in order to contain the hyperparameter space and runtime, while preventing overfitting. Finding a good trade–off within the model complexity is one of the most active research area in the GNN domain [129, 198].

Multiple attempts have been made in the graph kernel field to overcome this limitation, mainly addressing it from the perspective of an optimal assignment problem, i.e. to find good matching between substructure by minimizing a given cost function. Fröhlich *et al.* [57] proposed a kernel that performed optimal assignment on molecular graphs at the node label level. Nonetheless, it was later shown that this kernel is not positive definite [182], potentially creating inaccuracies when used as input for kernelized learning algorithms. Later on, Kriege *et al.* [105] developed a WL based kernel that employs optimal assignment on the node features at multiple iterations. However, these methods, as well as most of the existing graph kernels, have a major restriction in their applicability since they do not generalized on continuously attributed graphs. Extensions exist, mostly relying on hashing-based techniques, which can still result in a loss of information [126, 130]. We propose a method that overcomes the limitations of $\mathcal{R}$ – convolution kernels while being suitable for graphs with continuous attributes. This is achieved by integrating ideas from optimal transport theory, defining a kernel on vectorial graph representations obtained with a propagation scheme that iteratively aggregates information from the original, high–dimensional node attributes. Our so-

lution is built upon an efficient computation of Wasserstein distances [4, 36, 183] extended to the graph domain, paired with a Weisfeiler–Lehman inspired embedding scheme that can handle arbitrarily attributed and weighted graphs, at node and edge level. Our approach, the Wasserstein Weisfeiler–Lehman (WWL) graph kernel [172] has shown successful experimental performance on several benchmark data sets for graph classification, and in particular molecular graphs, outperforming the state–of–the–art.

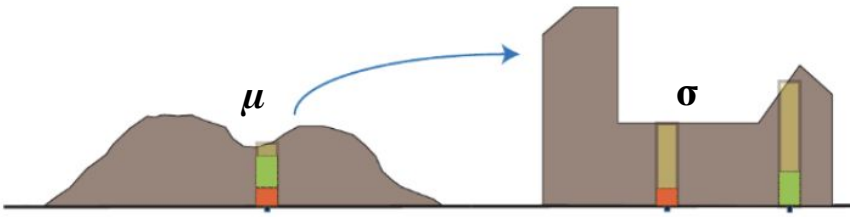
The remainder of this chapter is organised as follows. We begin by reviewing the basic concepts from optimal transport theory and define the Wasserstein distance, which is at the foundation of our method (section 5.1). Then, we extend the Wasserstein distance to graph structured data at the node feature level and derive a WL based propagation scheme to construct node embeddings (section 5.2). In section 5.3 we investigate how to obtain valid kernels from the Wasserstein distance on graphs. Experiments evaluating the empirical performance of our approach in comparison to the state–of–the–art are described in section 5.4, both with respect to classification accuracy and runtime. We conclude by summarizing our contributions and sketching ideas for future work (section 5.5).

5.1 Optimal transport

In this section we introduce concepts from optimal transport theory, particularly the Wasserstein distance as a measure of similarity, that will be later expanded to accommodate the graph setting.

Informally, the aim of optimal transport is to find the best matching or transportation that minimises the distance between two probability distributions. In other words, the is goal to find the functional minimum cost to transform a distribution into another one, where the objective minimising the cost is chosen accordingly to the problem of interest. The Wasserstein distance is a core ingredient in optimal transport, since it provides a distance measure between probability distributions, that can be later employed to optimize the cost in terms. More precisely, assuming we have samples or probability mass from two distributions σ and μ the Wasserstein distance can be interpreted as a ground distance between them. Then the optimal transport problem aims at solving the optimization to find the most "inexpensive" way to transform σ into μ. We can think about the optimal transport problem in a one–dimensional domain via an intuitive example. Suppose we have a pile of sand and we want to reassemble it to create a second pile of a different "shape" or to fill a hole; the optimal mass transportation problem aims at finding the minimal effort way to transform the pile from one distribution to the other one. An illustration of this example is given in Figure 5.1. Another typical intuitive representation of the optimal transport problem was suggested by Monge in his initial formulation [124]. Assume we have a set of bakeries producing bread every morning and having to deliver it to *cafés*, supposing that we know the amount of bread that will be consumed at each *café*. We can model this amount as a probability measure in a certain space, which here corresponds to the city map, equipped with a natural distance between points given by their shortest path (i.e. bakery-*café* distance). The goal is to

Figure 5.1: Optimal mass transportation problem. The sand pile on the left is assembled to build the pile on the right. Source: Mémoli [123].



find the best transport strategy by determining the amount of bread going from each bakery to each *café*, such that the transportation cost is minimised (Figure 5.2; [183]). The general idea of optimal transport problem was formulated by Monge [124] and later revisited by Kantorovich [96]. Then, the formulation as described in this section is also called the Kantorovich, or Monge-Kanterovich problem [183].

5.1.1 Wasserstein distance

To introduce the optimal transport problem from a more technical perspective, let us begin by recalling the notion of *coupling* in probability theory, of which the optimal transport plan is one of the most famous instances.

Definition 5.1 (Coupling [183])**.** We are given two probability distributions μ and σ in some space $\mathcal{X}$ and $\mathcal{Y}$, with random variables $X \sim \mu$ and $Y \sim \sigma$. Then, we say that (X, Y) is a coupling on (μ, σ) if X and Y are defined on a joint probability space $\Sigma = \mathcal{X} \times \mathcal{Y}$ and follow the same distribution as X and Y in the original space.

Equivalently, one can say that a measure π is defined on $\mathcal{X} \times \mathcal{Y}$ such that μ and σ are the *marginals* on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Optimal transport is an instance of coupling, i.e. a law between two distributions defined on a joint probability space.

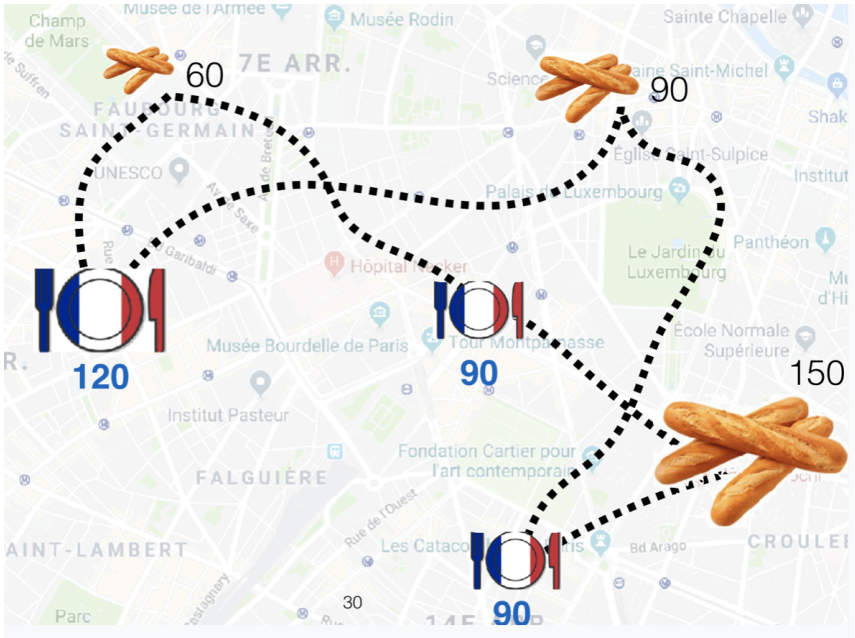
Let $(\mathcal{X}, \mu)$ and $(\mathcal{Y}, \sigma)$ be two probability spaces and define a cost function $c(x, y)$ on $(\mathcal{X} \times \mathcal{Y})$. According to the previous discussion, we interpret c as the cost of transforming one distribution into another one. The optimal transport minimization problem is formulated over all possible couple of random variables $X \sim \mu$ and $Y \sim \sigma$ to find

$$C(X, Y) = \inf \mathbb{E}[c(X, Y)], \tag{5.1}$$

or equivalently in terms of a probability measure

$$C(X, Y) = \inf \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\gamma(x, y). \tag{5.2}$$

Figure 5.2: Illustration of the optimal transport problem with the bakery example. Source: slides from M. Cuturi for a course on Computational Optimal Transport [142].



We minimise with respect to the set of all joint probability measures $\gamma \in \Gamma(\sigma, \mu)$, where $\Gamma(\sigma, \mu)$ are called *transport plans* and the solution of 5.2 is the *optimal transport plan*. The Wasserstein distance is defined as a special instance of equation 5.2.

Definition 5.2. The L^p-Wasserstein distance for $p \in [1, \infty)$ is given by

$$W_p(\sigma, \mu) := \left(\inf_{\gamma \in \Gamma(\sigma,\mu)} \int_{M \times M} d(x,y)^p \, \mathrm{d}\gamma(x,y) \right)^{\frac{1}{p}}, \tag{5.3}$$

where $\Gamma(\sigma, \mu)$ is the set of all transportation plans $\gamma \in \Gamma(\sigma, \mu)$ with marginals σ and μ, over $M \times M$, where M contains the distances $d(x, y)$.

Here, d is an arbitrary ground distance, for example Euclidean.

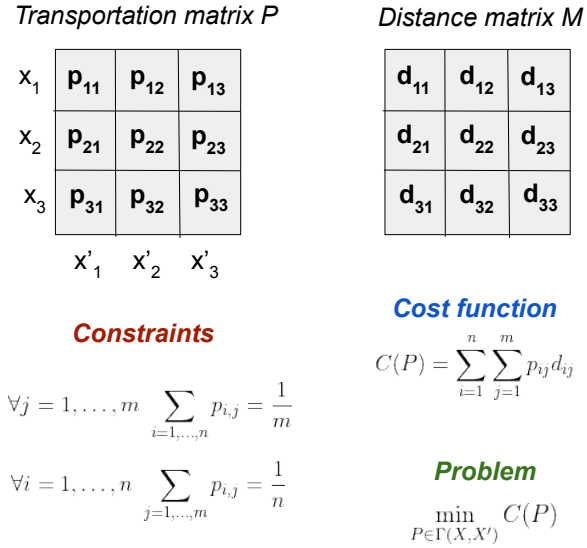
Theorem 5.1 (Villani [183])**.** *The Wasserstein distance satisfies the axioms of a metric, if d is a metric.*

Proof. The three properties of a metric can be checked as follows.

(i) $W_p(\sigma, \mu) = W_p(\mu, \sigma)$. It follows by definition, assuming that d is a metric then $d(x, y) = d(y, x)$.

(ii) If $\sigma = \mu$ then there exist X, Y with $X = Y$ and $d(X, Y) = 0$ and $W_p(\sigma, \mu) = 0$. Similarly, if $W_p(\sigma, \mu) = 0$ there exist the diagonal transport plan with $X = Y$ implying that $\mu = \sigma$.

Figure 5.3: Schematic view of the optimal transport problem in the discrete setting. Source: adapted from a slide of M. Cuturi for a course on Computational Optimal Transport [142].



(iii) Let $\sigma_1, \sigma_2, \sigma_3$ probability measures on $\mathcal{X}$, with X_1, X_2 and X_2, X_3 be the optimal plan for σ_1, σ_2 and σ_2, σ_3. Then, by the gluing lemma (see Villani [183]) there exist random variables X_1', X_2', X_3' such that $(X_1', X_2') \sim (X_1, X_2)$ and $(X_2', X_3') \sim (X_2, X_3)$ and X_1', X_3' is an optimal plan for σ_1, σ_3. Therefore, if d is a metric we can write:

$$
\begin{aligned}
W_p(\sigma_1, \sigma_3) &\leq (\mathbb{E} d(X_1', X_3')^p)^{\frac{1}{p}} && (5.4)\\
&\leq (\mathbb{E}(d(X_1', X_2')^p + d(X_2', X_3')^p))^{\frac{1}{p}} && (5.5)\\
&\leq (\mathbb{E} d(X_1', X_2')^p)^{\frac{1}{p}} + (\mathbb{E} d(X_2', X_3')^p)^{\frac{1}{p}} && (5.6)\\
&= W_p(\sigma_1, \sigma_2) + W_p(\sigma_2, \sigma_3) && (5.7)
\end{aligned}
$$

following from the Minkowski inequality and knowing that (X_1', X_2') and (X_2', X_3') are optimal plans.

□

A special case of the Wasserstein distance for $p = 1$ is also called L^1 – Wasserstein. We will mostly focus on this in the remainder of this chapter and refer to it as Wasserstein distance, unless specified otherwise. Our ultimate goal, is to extend the Wasserstein distance in the setting of node embeddings, in particular to evaluate the distance between set of nodes. Therefore, we can restrict our methodology to the discrete setting, in practice replacing the integral with a simpler sum and reformulate the problem in

matrix notation [153]. Let $X \in \mathbb{R}^{n \times p}$, $X' \in \mathbb{R}^{m \times p}$ the two set of vectors (e.g. node embeddings), then the Wasserstein distance can be written as:

$$W_1(X, X') := \min_{P \in \Gamma(X, X')} \langle P, M \rangle. \tag{5.8}$$

Here, $M \in \mathbb{R}^{n \times m}$ is a matrix containing all the distance values between vectors $d(x, x')$, for each pair of $x \in X$ and $x' \in X'$, while $P \in \Gamma$ is a transport matrix and $\langle \cdot, \cdot \rangle$ is the Frobenius dot product. The matrix $P \in \mathbb{R}^{n \times m}$ contains all the valid transport plans to transfer values from X to X', determining the fraction of mass to be transported. Because we are in a probabilistic framework, the total mass to be transferred from X to X' equals 1, it follows that the row and columns of P sum to $\frac{1}{n}$ and $\frac{1}{m}$ respectively. A schematic view is presented in Figure 5.3.

5.2 Wasserstein distance on graphs

In this section we introduce the major methodological contribution of our work, the Wasserstein distance on graphs and corresponding graph embedding scheme. We recall that our motivation to enhance graph kernel with optimal transport based distances, lies in the unsatisfactory nature of the $\mathcal{R}$ – convolution kernels. The averaging step might result in loss of substructure similarity, following our need to build more informative similarity measures that can account for complex interactions. Our method is developed upon 3 main steps:

1. Graph embedding scheme: transform each graph into a new representation as a set of node embeddings
2. Graph Wasserstein distance: evaluate the Wasserstein distance between graphs
3. Compute a similarity matrix from the distance and use it in the learning algorithm

We will now elucidate steps (1) and (2), while step (3) will be investigated in the next section.

5.2.1 Graph embedding scheme

The goal of the graph embedding scheme is to generate an accurate representation of each graph as a set of node embeddings.

Definition 5.3 (Graph Embedding Scheme)**.** Given a graph $G = (V, E)$, a graph embedding scheme $f \colon G \to \mathbb{R}^{|V| \times p}$, $f(G) = X_G$ is a function that outputs a fixed-size vectorial representation for each node in the graph. For each $v_i \in V$, the i-th row of X_G is called the node embedding of v_i.

A priori, Definition 5.3 does not make any assumption on f which can be an arbitrary function. The dimension p depends on the function f; typically, the node attributes or

categorical labels are used to determine the input dimension p. We will now present a graph embedding scheme inspired by the WL kernel, that is general enough to allow for either continuously attributed and categorically labels graphs, both as the edge and node level.

Node embeddings

We begin by recalling the WL iterative label scheme, as defined in section 2.2.5. With a similar notation let us define a graph $G = (V, E)$ with $\ell(v) = \ell^0(v)$ being the initial node label, for each node $v \in V$. For the moment, let us assume that the graph is categorically labelled, i.e. $\ell^0(v) \in \mathbb{N}$. Let H be the number of iteration of WL, then we defined the recursive scheme to generate node labels at different iterations as:

$$\ell^{h+1}(v) = hash(\ell^h(v), \mathcal{N}^h(v)). \tag{5.9}$$

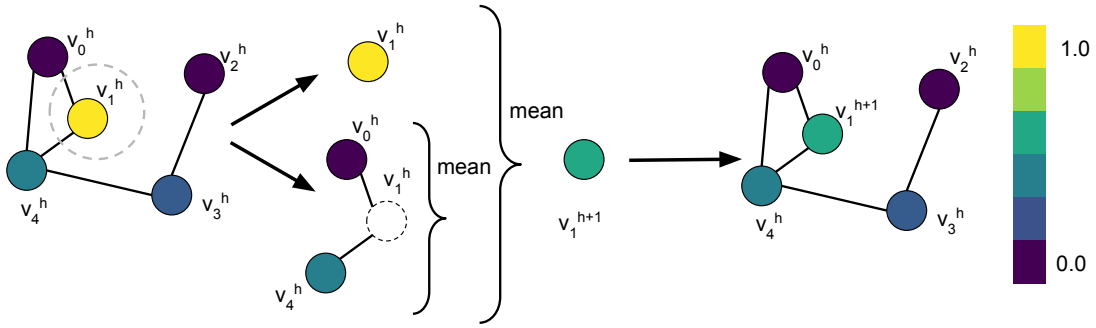
where $\mathcal{N}^h(v) = \{\ell^h(u_0), \dots, \ell^h(u_{deg(v)-1})\}$ is the neighbourhood of v, with neighbourhood node labels defined via WL at iterations h. Given the perfect hashing function, the updating rule might be too strict in terms of similarity, as the algorithm cannot distinguish between partially overlapping and totally different neighbours. Furthermore, the updating scheme as provided by the original WL does not extend to continuously attributed graph. We then modify the WL scheme to account for continuous attributes and partial similarities, to also resembles the updates step in GNNs. Nevertheless, contrarily to a GNN, our approach does not learn the updating function which is assumed to be fixed. While having the disadvantage of less flexibility, such an approach enjoys a tremendous speed up and is still powerful enough to detect hidden similarities, with a low risk of overfitting on a small sample size regime.

Continuous WL scheme. To make a clear enough distinction from the categorical case, we will now denoted the continuous attribute as $a(v) \in \mathbb{R}^p$, rather than $\ell(v)$. As before, $a^0(v) = a(v)$ denote the original node attributes for each $v \in V$. The idea behind the continuous WL scheme resemble the categorical case, by creating updates that leverage the information of the current node features and average over the neighbourhoods to create the updated embedding at the next iteration. Efforts in this direction have been already made to compute kernels that encode these node–level similarities on the continuous features. However, they usually rely on additional hashing or binarization of the continuous attributes, therefore loosing relevant information from the input [126, 130]. We define the recursive step to compute continuous WL features as follows:

$$a^{h+1}(v) = \frac{1}{2}\left(a^h(v) + \frac{1}{deg(v)} \sum_{u \in \mathcal{N}(v)} w((v,u)) \cdot a^h(u)\right). \tag{5.10}$$

A visual overview of our embedding scheme is provided in Figure 5.4. Here, $w(v,u)$ is the edge weight, if available, and $w(v,u) = 1$ otherwise. The term in parenthesis from Equation 5.10 is a sum between the node feature itself and a weighted average of

Figure 5.4: Intuitive representation of the WL graph embedding scheme. The node feature of the current node (v_1^h, yellow) at iteration h is updated by averaging the nodes in its neighbourhood (v_2^h, v_4^h), then again evaluating their mean. The new node feature at iteration $h+1$ is obtained v_1^{h+1}. This procedure is applied to every node in the graph.



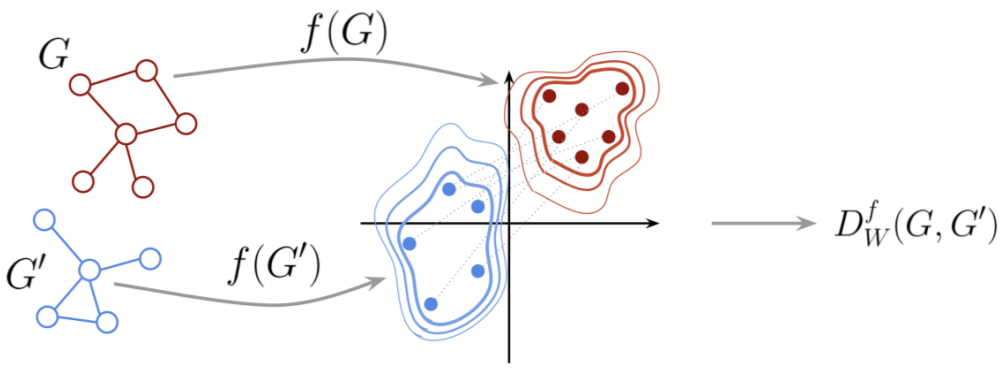
the neighbourhood node features, since $deg(v) = |\mathcal{N}(v)|$ and $w(v,u)$ is the weighting factor. For the neighbourhood component, it is also possible to use a customised or expert guided weight term w instead of the edge weights, though we believe that our current choice keeps the formulation as general as possible and applicable to any graph, without the need for extra domain knowledge. Additionally, we could also add another scaling factor on the node features themselves, though again an informed decision on this value should be guided by a learning or expert based criteria, which might not be known in the generic framework. The $1/2$ factor ensures a similar feature scale across iterations., while leveraging the contribution of the node itself and the neighbourhood. Indeed, due to the sum term, without a proper scaling the feature value itself would explode as the iteration number increases, leading to embeddings not fully comparable over the different iterations. Per our definition, the WL continuous refinement step is not directly related to a test of isomorphism as for the categorical setting; nevertheless it is a natural extension of it (see Sections 2.2.5 and 2.3).

WL FEATURES. Combining the universal graph embedding scheme with the WL refinement step for continuous and categorical label, we define a WL based graph embedding procedure. This generates the so–called *WL features* which can be interpreted as the node features obtained via the WL scheme; the input is given by either the continuous or categorical attributes.

Definition 5.4 (WL features)**.** Let $G = (V, E)$ and let H be the number of WL iterations. Then, for every $h \in \{0, \dots, H\}$, we define the WL features as

$$X_G^h = [x^h(v_1), \dots, x^h(v_{n_G})]^T, \tag{5.11}$$

Figure 5.5: Visual summary of the graph Wasserstein distance. First, f generates embeddings for two input graphs G and G'. Then, the Wasserstein distance between the embedding distributions is computed.



where $x^h(\cdot) = \ell^h(\cdot)$ for categorically labelled graphs and $x^h(\cdot) = a^h(\cdot)$ for continuously attributed graphs. We refer to $X_G^h \in \mathbb{R}^{n_G \times p}$ as the *node features* of graph G at iteration h. Then, the node embeddings of graph G at iteration H are defined as

$$\begin{aligned} f^H \colon G &\to \mathbb{R}^{n_G \times (p(H+1))} \\ G &\mapsto \text{concatenate}(X_G^0, \ldots, X_G^H). \end{aligned} \tag{5.12}$$

It is worth to mention that it is possible to also *jointly* consider continuous and categorical labels, for example by concatenating them. However, as we will see in the next section, we ultimately have to calculate a distance between node features to construct the Graph Wasserstein Distance. While it is easy to choose ground distances on either the continuous or categorical case, establish a joint measure is far from being trivial, and we leave this extension for future work [165].

5.2.2 Graph Wasserstein Distance

Once the node feature have been computed, for example with the WL embedding scheme, the next step is to compute a distance between those. We define a Wasserstein based distance between graphs as a distance between their node embeddings, and we will refer to it as Graph Wasserstein Distance (GWD).

Definition 5.5 (Graph Wasserstein Distance)**.** Assume we have two graphs $G = (V, E)$ and $G' = (V', E')$ and a graph embedding scheme $f \colon G \to \mathbb{R}^{|V| \times p}$ to output their node representation. Let $d \colon \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a ground distance defined on individual pair of node vectors. We define the Graph Wasserstein Distance (GWD) as

$$D_W^f(G, G') := W_1(f(G), f(G')). \tag{5.13}$$

The GWD should be interpreted as a measure of affinity between graphs, as represented by the set of their node embeddings. When the GWD runs over the entire graph, the embedding correspond to the random variable in Equation 5.8, so the objects are represented as a distribution of nodes. This characterisation allows to preserve partial similarities across node embeddings, assessing the closeness between their representative vectors. A visual summary of the first two steps of our method, consisting of the graph embedding scheme and computation of the Graph Wasserstein Distance is reproduced in Figure 5.5.

COMPUTING THE DISTANCE. The ground distance d in Definition 5.5 should be a valid metric. Different choices are possible depending on the nature of the embeddings, i.e. categorical versus continuous. For categorical node features, we use a normalised version of the Hamming distance:

$$d_{\text{Ham}}(v, v') = \frac{1}{H+1} \sum_{i=1}^{H+1} \rho(v_i, v'_i), \quad \rho(x, y) = \begin{cases} 1, & x \neq y \\ 0, & x = y \end{cases} \tag{5.14}$$

The Hamming distance is equivalent to a normalised sum, over the number of WL iterations H, of the discrete metric ρ evaluating the discrepancy between node features. If the vectors are identical, then the Hamming distance is 0, and if they have no common features their distance is 1. We observe that it is legitimate to use this distance assuming that the categorical node labels do not have any meaning per se, e.g. ordering, and they can be transformed to a one–hot–encoding fashion. In the classical WL, this is guaranteed by the neighbourhood aggregation and hashing step (see 2.2.5). In the continuous setting, we use a standard Euclidean distance between node features:

$$d_E(v, v') = ||v - v'||_2. \tag{5.15}$$

These distances, and in principle any other one appropriate for the problem of interest, should be plug-in to Equation 5.3 and the optimal transport problem is then solved with a network simplex method [142].

ALTERNATIVES TO THE WASSERSTEIN DISTANCE. While the Wasserstein Distance is appealing as providing a probabilistic perspective to the similarity score, it is imaginable to replace it with any other measure. A valuable alternative would be to use a Gaussian distance metric, i.e. replacing the GWD with an RBF kernel. Using a kernel instead of a distance also has the benefit to be a ready–to–use matrix for the learning algorithm, contrarily to the Wasserstein based approach where we need to take an extra step to compute the kernel (see Section 5.3). We define the RBF–WL as the approach employing an RBF kernel on the WL features.

Definition 5.6 (RBF–WL)**.** Let $G = (V, E)$ and $G' = (V', E')$ two graphs with $|V| = n$, $|V'| = n'$ with WL features at each iteration h given by X^h_G and $X^h_{G'}$, respectively. Then, we define the node kernel matrix as:

$$K^h(G, G') = RBF(X^h, X'^h) \tag{5.16}$$

with $K^h(G, G') \in \mathbb{R}^{n \times n'}$. Ultimately, we sum up the element of $K^h(G, G')$ to get:

$$K^h_{RBF-WL}(G, G') = \sum_{i=1}^{n} \sum_{j=1}^{n'} K^h(G, G')_{i,j} \tag{5.17}$$

as a kernel similarity value between G and G'.

The parameter h should be tuned in the learning algorithm. To get the final similarity matrix on a set of graphs, the kernel should be computed pairwise, at the cost of an high computational complexity. It is easy to see that such an approach is theoretically legitimate, as proved by the closure properties of kernels; indeed, any other valid kernel, could be used instead of the RBF.

5.3 From distance to kernels

We presented how the GWD results in a distance over graphs applicable to arbitrary node embeddings. However, for the use of classification and regression learning algorithms as kernel based methods, we need to go one step further and obtain a valid kernel from the distance matrix. Luckily, there are multiple ways to convert a distance matrix into a similarity measure, while certain conditions need to be verified for it to be a PSD kernel. In this section, we will show how to obtain kernels form the GWD and investigate on their (in)definiteness.

All the steps performed so far, grant us all the ingredients to define the final crucial contribution of our work, the Wasserstein–Weisfeiler Lehman kernel.

Definition 5.7 (Wasserstein Weisfeiler–Lehman)**.** Given a set of graphs $\mathcal{G} = \{G_1, \ldots, G_N\}$ and the GWD defined for each pair of graph on their WL embeddings, we define the Wasserstein Weisfeiler–Lehman (WWL) kernel as

$$K_{\text{WWL}} = e^{-\lambda D_W^{f_{\text{WL}}}}. \tag{5.18}$$

In general, a kernel defined as in Equation 5.18 for some ground distance belongs to the family of the *geodesic Laplacian kernel*. These kernels have been shown to provide theoretical guarantees for positive definiteness under favourable conditions, even for non-Euclidean distances, which is generally a trickier scenario to prove [49]. To investigate the theoretical properties of our kernel and assess the positive definiteness, we will distinguish between the continuous and categorical case, depending on the attribute nature of the graphs. The whole procedure consisting of WL features genera-

Algorithm 1 Compute Wasserstein graph kernel

Input: Two graphs G_1, G_2; graph embedding scheme f^H; ground distance d; λ.
Output: kernel value $k_{WWL}(G_1, G_2)$.
$X_{G_1} \leftarrow f^H(G_1)$; $X_{G_2} \leftarrow f^H(G_2)$ // Generate node embeddings
$D \leftarrow$ pairwise_dist(X_{G_1}, X_{G_2}, d) // Distance between pair of node embeddings
$D_W(G_1, G_2) = \min_{P \in \Gamma} \langle P, D \rangle$ // Compute the Wasserstein distance
$k_W(G_1, G_2) \leftarrow e^{-\lambda D_W(G_1, G_2)}$

tion, GWD and finally obtaining the WWL completes the methodological development or our approach, which is outlined in Algorithm 1.

5.3.1 Definiteness of the WWL

In the context of Euclidean space, it is well known how to generate valid kernels from distances, while theoretical and practical aspects have been widely investigated [78]. Unfortunately, the general Wasserstein distance does not generate a Euclidean space, i.e. it is not isometric to an L^2 norm and the corresponding metric space depends on the choice of the ground distance and type of input [51]. Overall, being a metric (e.g. Wasserstein distance) is a necessary but not sufficient condition to generate positive definite kernels with standard substitution approaches [78], motivating our need for a more in depth investigation on the subject. Several attempts have been made to establish the positive definiteness from optimal transport problems, and the field is still an active research area. Nevertheless, general considerations and results from the distance based and Laplacian application can be useful and extended to our particular setting.

Definition 5.8 (Conditional definite kernel)**.** Given a symmetric function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ yielding a positive definite kernel, i.e. $\sum_{i,j=1}^{n} c_i c_j K_{ij} \geq 0$, *with* $K_{ij} = k(x_i, x_j)$ for every $c_i \in \mathbb{R}$, $n \in \mathbb{N}$ and $x_i \in \mathcal{X}$, we say that k is *conditional* positive definite (CPD) if the condition holds for all $c_i \in \mathbb{R}$ with $\sum_{i=1}^{n} c_i = 0$. Analogously, if $\sum_{i,j=1}^{n} c_i c_j K_{ij} \leq 0$ for all $c_i \in \mathbb{R}$ with $\sum_{i=1}^{n} c_i = 0$, we say that k is *conditional* negative definite (CND).

The conditional positive definiteness is a weaker condition than classical PD, as it restricts its validity to a subset of the input space. Nevertheless, it is sometimes easier to generate CPD kernels from certain distance functions.

Proposition 5.2. *[78] Let $d(x, x')$ be a symmetric, non-negative distance function with $d(x, x) = 0$. If d is isometric to an L^2-norm, then*

$$k_d^{\text{nd}}(x, x') = -d(x, x')^\beta, \quad \beta \in [0, 2] \tag{5.19}$$

is a valid CPD kernel.

We refer to [78] for a proof. This proposition is extremely useful to define positive definiteness, presenting a very easy and intuitive way to transform a distance to kernel. However, we still have the issue that the Wasserstein distance, in its general formulation, is not isometric to an L^2 space. To overcome this limitation, Feragen *et al.* [49]

presented a family of exponential kernels that enjoy positive definiteness, under certain conditions, even when the distance function is non-Euclidean:

$$k(x, x') = e^{-\lambda d(x,x')^q} \quad \text{for} \quad \lambda, q > 0. \tag{5.20}$$

Proposition 5.3. *[49] Let us call the geodesic Laplacian kernel the one obtained from Equation 5.20 for $q = 1$. The geodesic Laplacian kernel is positive definite for all $\lambda > 0$ if and only if the geodesic distance d is conditional negative definite.*

We refer the interested reader to [49] for a complete poof, based on previous arguments and background presented in [15]. Despite this proposition, in the general formulation we still cannot guarantee negative definiteness of the Wasserstein distance.

The case of categorical embeddings

When the original node labels are categorical, the node embeddings will also be integers, in practice a concatenation of histograms. We will show that this condition, together with the WL procedure scheme, is enough to guarantee positive definiteness for the Laplacian based WWL kernel from Definition 5.7. At the core of our analysis is the following statement: if the Wasserstein distance is defined with a discrete metric as ground distance (e.g. the Hamming distance), then it is conditional negative definite [61]. Several considerations will then lead us to prove our final results concerning the positive definiteness of categorical WWL.

We begin by observing that the solutions to the optimal transport problem over node embeddings generated with the Weisfeiler–Lehman labelling scheme are also shared across iterations, since the label dictionary is shared within graphs. We denote the Weisfeiler–Lehman embedding scheme as defined in Definition 5.4 as f_{WL}^H, and let $D_W^{f_{\text{WL}}}$ be the corresponding GWD on a set of graphs $\mathcal{G}$ with categorical labels. Let $d_{\text{Ham}}(v, v')$ of Equation 5.14 be the ground distance of $D_W^{f_{\text{WL}}}$. Then, we can prove a series of useful results.

Lemma 5.4. *If a transportation plan γ with transport matrix P is optimal as established by Definition 5.2 for distances d_{Ham} between embeddings obtained with f_{WL}^H, then it is also optimal for the discrete distances d_{disc} between the H-th iteration values obtained with the Weisfeiler–Lehman scheme.*

Proof. Recalling the notation from Section 5.1.1 and in particular from 5.8, we denote by M the cost or distance matrix, also $P \in \Gamma$ is a transport matrix (or joint probability), and $\langle \cdot, \cdot \rangle$ is the Frobenius dot product. Since each of the vectors has equal weight (i.e., equal probability mass), Γ contains all nonnegative $n \times n'$ matrices P with

$$\sum_{i=1}^{n} p_{ij} = \frac{1}{n'} \ , \ \sum_{j=1}^{n'} p_{ij} = \frac{1}{n} \ , \ p_{ij} \geq 0 \ \ \forall i, j.$$

For notation simplicity, let us denote the Hamming matrix $D_{\text{Ham}}(f_{\text{WL}}^h(G), f_{\text{WL}}^h(G'))$ by D_{Ham}^h, where the ij-th entry is given by the Hamming distance between the embedding

of the i-th node of graph G and the embedding of the j-th node of graph G' at iteration h. Similarly, D^h_{disc} is defined to be the discrete metric distance matrix, with the ij-th entry is given by the discrete distance between feature h of node embedding i of graph G and feature h of node embedding j of graph G'. Therefore, the two matrices $[D^h_{\text{Ham}}]_{ij} \in [0,1]$ and $[D^h_{\text{disc}}]_{ij} \in \{0,1\}$ are restricted to values in the corresponding range and, by definition of the WL scheme, we obtain:

$$D^H_{\text{Ham}} = \frac{1}{H} \sum_{h=0}^{H} D^h_{\text{disc}}.$$

Additionally, from the classical WL procedure, it follow that if two labels are different at iteration h they will also be different at iteration $h+1$. Hence, we deduce that

$$\left[D^h_{\text{Ham}}\right]_{ij} \leq \left[D^h_{\text{disc}}\right]_{ij}$$

and consequently $[D^h_{\text{Ham}}]_{ij} = 0 \iff [D^h_{\text{disc}}]_{ij} = 0$. From the definition of an optimal transportation plan P^h for the embeddings f^h_{WL}, it always holds that:

$$\left\langle P^h, D^h_{\text{Ham}} \right\rangle \leq \left\langle P, D^h_{\text{Ham}} \right\rangle \; \forall P \in \Gamma.$$

Now, assume that P^h is not optimal for D^h_d. Then, there exists P^* such that

$$\left\langle P^*, D^h_{\text{disc}} \right\rangle < \left\langle P^h, D^h_{\text{disc}} \right\rangle.$$

Since we showed that the entries of D^h_{disc} are restricted to be either 0 or 1, we can define the set of indices tuples $\mathcal{H} = \left\{(i,j) \mid [D^h_{\text{disc}}]_{ij} = 1\right\}$ and rewrite the last inequality as:

$$\sum_{i,j \in \mathcal{H}} p^*_{ij} < \sum_{i,j \in \mathcal{H}} p^h_{ij}.$$

Again, the constraints on the entry values of P^* and P^h imply that $\sum_{i,j} p^*_{ij} = \sum_{i,j} p^h_{ij} = 1$ and, by rearranging the transport map, there is more mass that could be transported at 0 cost, i.e.

$$\sum_{i,j \notin \mathcal{H}} p^*_{ij} > \sum_{i,j \notin \mathcal{H}} p^h_{ij}.$$

However, as we observed, entries of D^h_d that are 0 must also be 0 in D^h_{Ham}. Therefore, a better transport plan P^* would also be optimal for D^h_{Ham}:

$$\left\langle P^*, D^h_{\text{Ham}} \right\rangle < \left\langle P^h, D^h_{\text{Ham}} \right\rangle.$$

This is a contradiction of the optimality assumption, then we can conclude that P^h is also optimal for D^H_{disc}. □

Lemma 5.5. *If a transportation plan γ with transport matrix P is optimal in the sense of Definition 5.2 for distances d_{Ham} between embeddings obtained with f_{WL}^{H}, then it is also optimal for distances d_{Ham} between embeddings obtained with f_{WL}^{H-1}.*

Proof. Intuitively, the transportation plan at iteration h is a "refinement" of the transportation plan at iteration $h-1$, where only a subset of the optimal transportation plans remains optimal for the new cost matrix D_H^h. As a consequence of Lemma 5.4 and in light of the WL procedure, two labels that are different at iteration h will also be different at iteration $h+1$. Applying the distances definition, the following inequalities can be obtained:

$$\left[D_{\text{Ham}}^{h}\right]_{ij} \leq \left[D_{\text{Ham}}^{h+1}\right]_{ij}$$
$$\left[D_{\text{disc}}^{h}\right]_{ij} \leq \left[D_{\text{disc}}^{h+1}\right]_{ij}$$
$$\left[D_{\text{Ham}}^{h}\right]_{ij} \leq \left[D_{\text{disc}}^{h}\right]_{ij}.$$

Also, an optimal transportation plan P^h for the WL embeddings $f_{WL}^{h}(G)$ has to satisfy

$$\left\langle P^h, D_{\text{Ham}}^{h} \right\rangle \leq \left\langle P, D_{\text{Ham}}^{h} \right\rangle \; \forall P \in \Gamma,$$

which is equivalent to

$$\left\langle P^h, D_{\text{Ham}}^{h} \right\rangle = \frac{1}{h}\Big((h-1)\cdot\left\langle P^h, D_{\text{Ham}}^{h-1} \right\rangle + \left\langle P^h, D_{\text{disc}}^{h} \right\rangle\Big).$$

We know that for increasing h, the values of D_{Ham}^{h} increase in a step–wise fashion and their ordering remains constant, except for entries that were 0 at iteration $h-1$ and became $\frac{1}{h}$ at iteration h. Given the monotonicity conditions of our metric, and since P^h is optimal for D_{disc}^{h}, from Lemma 5.4 we deduce that

$$\left\langle P^h, D_{\text{Ham}}^{h-1} \right\rangle \leq \left\langle P, D_{\text{Ham}}^{h-1} \right\rangle \; \forall P \in \Gamma.$$

Therefore, P^h is also optimal for the WL embeddings $f_{\text{WL}}^{h-1}(G)$ at iteration $h-1$. □

These results lead us to postulate that the Wasserstein distance between categorical WL node embeddings is a conditional negative definite function.

Theorem 5.6. *$D_W^{f_{\text{WL}}}(\cdot,\cdot)$ is a conditional negative definite function.*

Proof. Using the same notation as for Lemma 5.4, we obtain

$$\begin{aligned} D_W^{f_{\text{WL}}}(G, G') &= \min_{P^H \in \Gamma} \left\langle P^H, D_{\text{Ham}}^{H} \right\rangle \\ &= \min_{P^H \in \Gamma} \frac{1}{H} \sum_{h=0}^{H} \langle P^H, D_{\text{disc}}^{h} \rangle. \end{aligned}$$

Let P^* be an optimal solution for iteration H. Then, from Lemmas 5.4 and 5.5, it is also an optimal solution for D^H_{disc} and for all $h = 0, \ldots, H-1$. We can express this condition as a sum of optimal transport problems:

$$D_W^{f_{\text{WL}}}(G, G') = \frac{1}{H} \sum_{h=0}^{H} \min_{P^* \in \Gamma} \langle P^*, D^h_{\text{disc}} \rangle. \tag{5.21}$$

This corresponds to a sum of one–dimensional optimal transport problems relying on the discrete metric, which were shown to be conditional negative functions [61]. It follows that the final sum is also conditional negative definite. □

We are now in the position to state our main result for the definiteness of kernels in the categorical setting.

Theorem 5.7. *The categorical WWL kernel is positive definite for all $\lambda > 0$.*

Proof. The result is a direct consequence of Theorem 5.6 and Proposition 5.3. □

The case of continuous embeddings

In the continuous setting, it is still an open problem to determine the positive definiteness of our method. We postulate a series of considerations and we conjecture that, under particular conditions, it is possible to prove that WWL is PD for continuous embeddings as well. Nevertheless, we do no have a formal proof yet, which we leave as an extension for future work. We will now present several supporting arguments that further agree with our empirical findings (see Section 5.4). Indeed, we will observe that in our data sets, after standardisation of the input features before the embedding scheme, GWD matrices are conditional negative definite.

At an high level, our argument is based on the properties and *curvature* of metric spaces, and how they relate to the Euclidean space. The curvature is a concept in geometry determining the "shape" of a manifold, as well as their orientation. Euclidean spaces are "flat" and the curvature of a space is an indication of how much they "deviate" from the flatness. More formally, the curvature indicates to what extent a geodesic triangle will be deformed in the space. Determining the curvature ultimately characterizes the corresponding space.

Definition 5.9 (Alexandrov space)**.** A metric space is called an Alexandrov space if its sectional curvature is $\geq k$, for some real value k.

Flat spaces are characterised by a curvature of $k = 0$. Feragen *et al.* [49] shows that there is a strong connection between a kernel positive definiteness and the underlying metric space via its curvature, as an indication of its "closeness" to Euclidean spaces.

Proposition 5.8. *The geodesic Gaussian kernel (i.e., $q = 2$ in Equation 5.20) is positive definite for all $\lambda > 0$ if and only if the underlying metric space (X, d) is flat in the sense of Alexandrov, i.e., if any geodesic triangle in X can be isometrically embedded in a Euclidean space.*

It is generally not true that the space induced by the Wasserstein distance is (locally) flat, as not even the geodesics (i.e., shortest paths between points in metric spaces) connecting graph embeddings are unique, which is a requirement for a space to be flat. As for the categorical case, according to Proposition 5.3, we would need to prove that the metric used in the kernel function is CND. This is yet an open problem, but we can prove that the opposite is not true. In particular, if X is the is metric space induced by the GWD, we can show that its curvature is *not* bounded from above.

Definition 5.10. A metric space (X, d) is said to be CAT(k) if its curvature is bounded by some real number $k > 0$ from above.

Theorem 5.9. *X is not in* CAT(k) *for any $k > 0$, meaning that its curvature is* not *bounded by any $k > 0$ from above.*

Proof. Our argument is similar to the one presented in Turner *et al.* [174]. We provide here a sketch of the proof.

Given two graph G and G', let us assume that X is a CAT(k) space for some $k > 0$. Then, it has been shown [25, Proposition 2.11, p. 23] that if $D_W^{f_{\text{WL}}}(G, G') < \pi^2/k$, there is a *unique* geodesic between them. We observe that it is possible to construct a family of graph embeddings for which this is not the case. In particular, let $\epsilon > 0$ and $f_{\text{WL}}(G)$ and $f_{\text{WL}}(G')$ be two graphs with node embeddings $a_1 = (0, 0)$, $a_2 = (\epsilon, \epsilon)$ as well as $b_1 = (0, \epsilon)$ and $b_2 = (\epsilon, 0)$, respectively. Since we used the Euclidean distance as a ground distance, there exist two optimal transport plans: the first maps a_1 to b_1 and a_2 to b_2, whereas the second maps a_1 to b_2 and a_2 to b_1. Hence, we have found two distinct geodesics that connect G and G'. Choosing ϵ arbitrarily small, it follow that the space cannot be CAT(k) for $k > 0$. □

While this does not provide an upper bound on the curvature, we can state the the following conjecture.

Conjecture 5.10. *X is an Alexandrov space with curvature bounded from below by zero.*

For a proof idea, we refer to Turner *et al.* [174]; the main argument involves characterizing the distance between triplets of graph embeddings. The importance of this conjecture is hidden in the non–negativity of the curvature of Alexandrov spaces, a necessary condition for X to be a Hilbert space [136]. Furthermore, Feragen *et al.* [49] showed that CND metrics and Hilbert spaces are intricately linked, strongly indicating that such metrics could be obtained in our setting, under appropriate conditions. Moreover, our empirical results (Section 5.4), will indicate that it is possible to turn the GWD into a CND metric, after normalisation. The intuition is simple, as for high-dimensional input spaces, standardisation of input features changes the curvature of the induced space by making it locally (nearly) flat.

Indeed, arguments related to the flatness of the space have already been proposed as possible ways to ensure positive definiteness. For example, one can use an alternative to the classical Wasserstein distance denoted as the sliced Wasserstein [145]. The idea is

to project high–dimensional distributions into one–dimensional spaces, hereby calculating the Wasserstein distance as a combination of these representations. Kolouri *et al.* [101] showed that in one dimension each of the Wasserstein distances is CND, therefore obtained a kernel on high-dimensional representations as a combination of the one–dimensional positive definite counterparts.

5.3.2 Kreĭn Support Vector Machines

So far, our considerations cannot guarantee the positive definiteness of the WWL in the general continuous case. We empirically observed that our kernel matrices are nearly positive definite (see Section 5.4), meaning that their eigenvalues are close to zero. While the kernelized SVM is robust to these cases, we still aim to guarantee the theoretical correctness of our approach. In fact, the positive definiteness of the kernel matrix is a necessary condition to ensure exactness of the kernelized SVM (Equation 3.9). Therefore, we employ a new class of algorithms recently developed that formally extends the SVM for learning with indefinite kernels [138]. It has been shown that for non positive Gram matrices one can define an extension of the RKHS, the reproducing kernel Kreĭn spaces (RKKS) induced by the kernel k, that shares many properties with the RKHS learning framework. The essential difference between RKHS and RKKS is in the characterization of the inner product, which is indefinite in the Kreĭn space.

Definition 5.11 (Kreĭn space [115, 138])**.** Let $(\mathcal{K}, \langle \cdot, \cdot \rangle)$ be a an inner product space. We say that $\mathcal{K}$ is a Kreĭn space if there exist two Hilbert spaces $\mathcal{H}_+, \mathcal{H}_-$ spanning it, such that:

1. For each $f \in \mathcal{K}$ there is a decomposition $f = f_+ + f_-$, with $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$
2. For all $f, g \in \mathcal{K}$ the inner product can be written as:

$$\langle f, g \rangle_{\mathcal{K}} = \langle f_+, g_+ \rangle_{\mathcal{H}_+} - \langle f_-, g_- \rangle_{\mathcal{H}_-}$$

This definition implies that there exist an associated Hilbert space $\mathcal{H}$ and if both $\mathcal{H}_+, \mathcal{H}_-$ are RKHS, then we say that $\mathcal{K}$ is a reproducing kernel Kreĭn space. For a complete characterization, we refer the interested reader to previous literature [9, 18, 138]. When extending the SVM to RKKS the issue is in the optimization problem, since the lack of a positive inner product implies that the loss in the dual problem can no longer be minimized (Equation 3.8). The solution relies on replacing the minimisation problem with a stabilization approach, which has been shown to provide theoretical correctness and good empirical results [138]. This approach has been later proved to be a valuable solution to solve the SVM problem and extended for applicability in this domain [115]. Alternatives to the standard stabilization technique have been proposed, which employ regularization to improve theoretical consistency and guarantee, while providing an efficient and effective solution [135]. Besides the theoretical relevance, these approaches showed clear benefits from learning in RKKS when the kernel is not guaranteed to be PSD, also in terms of classification performance. Therefore, in our experiments we use a Kreĭn SVM (KSVM, [115]) as a classifier for the case of continuous attributes.

Table 5.1: Description of the experimental data sets.

Data set	Class Ratio	Node Labels	Node Attributes	Edge Weights	# Graphs	Classes
MUTAG	63/125	✓	-	-	188	2
NCI1	2053/2057	✓	-	-	4110	2
PTC-MR	152/192	✓	-	-	344	2
D&D	487/691	✓	-	-	1178	2
ENZYMES	100 per class	✓	✓	-	600	6
PROTEINS	450/663	✓	✓	-	1113	2
BZR	86/319	✓	✓	-	405	2
COX2	102/365	✓	✓	-	467	2
SYNTHIE	100 per class	-	✓	-	400	4
IMDB-B	500/500	-	(✓)	-	1000	2
SYNTHETIC-NEW	150/150	-	✓	-	300	2
BZR-MD	149/157	✓	-	✓	306	2
COX2-MD	148/155	✓	-	✓	303	2

5.4 Experiments

We now analyse the classification performance of WWL in comparison to state–of–the–art graph kernels on molecular and collaboration graph datasets, with respect to several tasks. We also perform a runtime experiment to investigate the computational requirement of our approach, and assess the benefit of applying speed-up approximation tricks [4, 36].

5.4.1 Data sets

To investigate the classification performance we collect multiple real–world and synthetic benchmark data sets from the graph kernel literature [161, 184]. Complete information of the data is reported in Table 5.1. All data sets have been downloaded from a public repository[1] [99].

Data description. The data sets are equipped with either continuous or categorical attributes, and therefore suitable for both variants of WWL. Some of the data sets contain categorical labels only, namely MUTAG, PTC-MR, NCI1, and D&D; others, have both categorical and continuous attributes (ENZYMES and PROTEINS); additionally, IMDB-B, BZR, COX2, SYNTHIE and SYNTHETIC-NEW only have continuous attributes; finally, BZR-MD and COX2-MD contain both node labels and edge weights. Most of these data sets belong to the chemoinformatics domain and include small molecules (MUTAG, PTC-MR, NCI1), macromolecules (ENZYMES, PROTEINS; [22]) and chemical compounds (BZR, COX2; [169]). We also include two sythentic data sets SYNTHIE and SYNTHETIC-NEW, created by Morris *et al.* [126] and

[1] https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

Feragen *et al.* [50], respectively. Lastly, a movie collaboration data set IMDB [200] is also considered.

DATA PROCESSING. The BZR-MD and COX2-MD data sets are the only ones containing an edge weight, i.e. the atomic distance between each connected atom. Here, we only consider distances between connected nodes [102] as edges and to obtain a node attribute we use the one–hot–encoding of the original node labels, representing the atom type. For IMDB-B we employed the node degree as a (semi-)continuous node feature [200]. For all the other data sets we use the off–the–shelf version provided in the repository [99].

5.4.2 EXPERIMENTAL SETUP

To assess the learning capability of WWL as a kernel similarity measure we compare its prediction power with well established graph kernels. Referring to the methods description in Section 2.2, the following approaches are included as competitors in the categorical setting: Weisfeiler-Lehman subtree kernel (WL); Weisfeiler-Lehman optimal assignment kernel (WL-OA); node histogram kernel (NH); edge histogram kernel (EH). For continuously attributed graphs, we compare WWL with the graph hopper kernel (GH) and two variants of the hash graph kernel (HGK-SP; HGK-SP); we also consider two baselines directly derived by node embeddings, the all node-pairs kernel with an RBF as base kernel (N-RBF), and the RBF-WL from Definition 5.6.

Either a KSVM or an SVM are used as classifiers, for the continuous and categorical setting, respectively. To evaluate the classifier generalisation ability we use a 10 -fold cross–validation, selecting the hyperparameters on the training set only. Each cross–validation split is repeated 10 times to account for randomness and we report the average accuracy and standard deviation. The same splits are used across all methods, in order to ensure comparability of our findings.

As for the classifiers hyperparameter tuning, we select the C of KSVM in the range $C = \{10^{-3}, \dots, 10^{3}\}$, for continuous attributes; in the categorical case, C of the SVM is chosen in the range $C = \{10^{-4}, \dots, 10^{5}\}$. For the kernel parameters, the number of iterations h of WL is selected in in the grid $h = \{0, \dots, 7\}$, while for the λ parameter of the WWL we use $\lambda = \{10^{-4}, \dots, 10^{1}\}$. For RBF-WL and N-RBF, we use the default γ parameter for the Gaussian kernel, that is we set $\gamma = 1/p$, where p is the size of node attributes. Following the recommendations in Feragen *et al.* [50] and Morris *et al.* [126], we also fix the γ parameter to $1/p$ in GH, and the number of iterations to 20 for each data set, except for SYNTHETIC-NEW where we use 100, for the HGK methods. Moreover, since HGK is a randomised method, each kernel matrix is computed 10 times and we average the results to get the final prediction score. For high–dimensional continuous attributes $p > 1$, these are normalised to ensure comparability among the different feature scales, in each data set except for BZR and COX2, where the node attributes are location coordinates hereby the normalisation would result in the loss of attributes meaning.

Table 5.2: Classification accuracies on graphs with categorical node labels. Comparison of Weisfeiler–Lehman kernel (WL), optimal assignment kernel (WL-OA), and our method (WWL). The best result is highlighted in bold. An * denotes a statistically significant difference between the best performing method and all other approaches.

Method	MUTAG	PTC-MR	NCI1	PROTEINS	D&D	ENZYMES
NH	85.39 ± 0.73	$58.35 \pm 0.20^*$	64.22 ± 0.11	72.12 ± 0.19	78.24 ± 0.28	22.72 ± 0.56
EH	84.17 ± 1.44	$55.82 \pm 0.00^*$	63.57 ± 0.12	72.18 ± 0.42	75.49 ± 0.21	21.87 ± 0.64
WL	85.78 ± 0.83	$61.21 \pm 2.28^*$	85.83 ± 0.09	74.99 ± 0.28	78.29 ± 0.30	53.33 ± 0.93
WL-OA	87.15 ± 1.82	$60.58 \pm 1.35^*$	$\mathbf{86.08 \pm 0.27}$	$\mathbf{76.37 \pm 0.30^*}$	79.15 ± 0.33	58.97 ± 0.82
WWL	$\mathbf{87.27 \pm 1.50}$	$\mathbf{66.31 \pm 1.21^*}$	85.75 ± 0.25	74.28 ± 0.56	$\mathbf{79.69 \pm 0.50}$	$\mathbf{59.13 \pm 0.80}$

When available, we use the implementation provided by the authors to compute the kernel, i.e. for HGK, WL-OA and GH.

5.4.3 Classification results

We evaluate the classification results in terms of accuracy for all the data sets, both in the categorical and continuous case. To evaluate the difference across methods and measure statistical significance, we perform a 2-sample t-tests with a threshold of 0.05 and Bonferroni correction for multiple hypothesis testing, within each data set. Nevertheless, since no meaningful comparison can be performed between the two settings, due to methods being not applicable in either of the scenario or the information used to compute the kernel (categorical or continuous) being different, we make a separate discussion between the two cases.

Categorically labelled graphs. The results of the categorical WWL against the competitor approaches are reported in Table 5.2. The main take home message is that, on the categorical data sets, WWL is comparable to the WL-OA kernel, and both WWL and WL-OA improve over the classical WL. Indeed, we see that on two data sets, PTC-MR and D&D, WWL is either clearly or slightly better, while on NCI1 and PROTEINS, WL-OA outperforms WWL. In the the other data sets, on MUTAG and ENZYMES the standard deviations overlap and no clear winning method can be identified. We conclude that the two approaches are comparable on these data sets. Such an observation does not come as a surprise, indeed the WL–OA formulation relies on solving the optimal assignment problem by defining Dirac kernels on histograms of node labels, using multiple iterations of WL. It is evident that this is very similar to WWL on categorical data, despite WL–OA relying on optimal assignment rather than the optimal transport; therefore, it requires one-to-one mappings instead of continuous transport maps. Another difference is that we solve the optimal transport problem on the concatenated embeddings, hereby jointly exploiting representations at multiple WL iterations. Contrarily, the WL–OA performs an optimal assignment at each iteration of WL and only combines them in the second stage. While these modifications result in different kernel

Table 5.3: Classification accuracies on graphs with continuous node and/or edge attributes. Comparison of hash graph kernel (HGK-WL, HGK-SP), GraphHopper kernel (GH), and our method (WWL). The best result is highlighted in bold. An * denotes a statistically significant difference between the best performing method and all other approaches

Method	ENZYMES	PROTEINS	IMDB-B	BZR	COX2	BZR-MD	COX2-MD
N-RBF	47.15 ± 0.79	60.79 ± 0.12	71.64 ± 0.49	74.82 ± 2.13	48.51 ± 0.63	66.58 ± 0.97	64.89 ± 1.06
RBF-WL	68.43 ± 1.47	75.43 ± 0.28	72.06 ± 0.34	80.96 ± 1.67	75.45 ± 1.53	69.13 ± 1.27	71.83 ± 1.61
HGK-WL	63.04 ± 0.65	75.93 ± 0.17	73.12 ± 0.40	78.59 ± 0.63	$\mathbf{78.13 \pm 0.45}$	$\mathbf{68.94 \pm 0.65}$	74.61 ± 1.74
HGK-SP	66.36 ± 0.37	75.78 ± 0.17	73.06 ± 0.27	76.42 ± 0.72	72.57 ± 1.18	66.17 ± 1.05	68.52 ± 1.00
GH	65.65 ± 0.80	74.78 ± 0.29	72.35 ± 0.55	76.49 ± 0.99	76.41 ± 1.39	$\mathbf{69.14 \pm 2.08}$	66.20 ± 1.05
WWL	$\mathbf{73.25 \pm 0.87}^*$	$\mathbf{77.91 \pm 0.80}^*$	$\mathbf{74.37 \pm 0.83}^*$	$\mathbf{84.42 \pm 2.03}^*$	$\mathbf{78.29 \pm 0.47}$	$\mathbf{69.76 \pm 0.94}$	$\mathbf{76.33 \pm 1.02}$

matrices, the difference is not pronounced enough to appreciate variations in performance on these data sets, possibly due to the type of labels or small sample size.

Nevertheless, the key advantage and empirical superiority of WWL over WL–OA is in its capacity to handle the continuous attributes, as we will discuss next.

CONTINUOUSLY ATTRIBUTED GRAPHS. Results on continuously attributed graphs on the real-world datasets are reported in Table 5.3. We observe that overall WWL shows high accuracies in comparison to the other approaches. In particular, on 4 datasets (ENZYMES, PROTEINS, IMDB-B, and BZR) WWL significantly outperforms the other methods. Specifically, on COX2 the performance is on par to HGK-WL, as both mean and standard deviation overlap. On the other 2 data sets, BZR-MD and COX2-MD, WWL still has the best performance, despite the difference with the second best approach being non statistically significant. Calculating the average rank of the methods, we obtain: WWL = 1, HGK-WL = 2.86, RBF-WL = 3.29, HGK-SP = 4.14, and VH-C = 5.86. From these perspective, WWL clearly scores as first, establishing a new state–of–the–art in graph kernels classification on the continuous setting. The HGK-WL method appears to be the strongest competitor of our approach. In spirit, the idea of HGK-WL is not so different from WWL, since it is based on a WL inspired propagation scheme to update the node features at each iteration. However, the HGK performs an hashing step to deal with the continuous case, compressing and potentially loosing the original information. To leverage for this issue, they use multiple hashing functions instead of a perfect map, but that might still not capture all the small differences between continuous attributes. Furthermore, the random hashing step requires additional hyperparameters making the generation of the kernel matrix sensitive to the chosen seed, while also increasing the runtime. The benefit of a fully continuous and deterministic approach is confirmed by the gap in performance observed between the WWL and HGK-WL. Our method, always outperforms the baselines with the RBF kernel. This result is particularly important in the context of the GWD component, implying that standard kernel between node features, such as the RBF, are not expressive enough to capture all the

Table 5.4: Classification accuracies on synthetic graphs with continuous node attributes. Comparison of hash graph kernel (HGK-WL, HGK-SP), GraphHopper kernel (GH), and our method (WWL).

Method	SYNTHIE	SYNTHETIC-NEW
N-RBF	27.51 ± 0.00	60.60 ± 1.60
RBF-WL	94.43 ± 0.55	86.37 ± 1.37
HGK-WL	81.94 ± 0.40	$\mathbf{95.96 \pm 0.25}^*$
HGK-SP	85.82 ± 0.28	80.43 ± 0.71
GH	83.73 ± 0.81	88.83 ± 1.42
WWL	$\mathbf{96.04 \pm 0.48}^*$	86.77 ± 0.98

hidden patterns and similarities between nodes. In the optimal transport formulation the node distribution is used, generating a more precise matching score between set of nodes.

We also perform a comparison on the synthetic data sets, as shown in Table 5.4. We observe that on these data sets the methods seem to give unstable results, with a large gap in performance across them. In our preliminary experiments, contrarily to the real world scenario, we observed that varying the scale of the node features (e.g., normalisation or scaling of the embedding scheme) resulted in a large change of performance (up to 15%). It is also unclear if the use of synthetic continuous node attributes is beneficial on these data sets, since both Morris *et al.* [126] and Feragen *et al.* [50] showed that on SYNTHETIC-NEW, the WL kernel with degree treated as categorical node label outperforms the competitors. Such considerations remark the crucial importance of choosing appropriate data sets for graph kernels when designing a novel approach, to ensure a fair assessment and comparison with the state–of–the–art.

POSITIVE DEFINITENESS. From a theoretical perspective, we pointed out that the kernel obtained from continuous attributes is not necessarily positive definite. Nevertheless, the empirical results supported our theoretical considerations, since the obtained kernel matrices are nearly positive definite, i.e. the eigenvalues are closed to zero. This suggests that after proper normalization the node feature space is (locally) nearly flat and, within certain bounds, properties of the Euclidean space still hold. Indeed, the difference between the results obtained from classical SVMs in RKHS and those with the KSVM approach is negligible, suggesting that in practice we might not need to account for the indefiniteness of the kernel.

5.4.4 RUNTIME AND COMPLEXITY

One of the disadvantages of the Wasserstein distance is the computational complexity, which is $\mathcal{O}(n^3 log(n))$, with n being the cardinality of the indexed set of node embed-

dings, i.e., the number of nodes in the two graphs. From a theoretical perspective, both WL and WL–OA scale linearly with the number of nodes, so they are computationally less expensive than WWL. Strategies to speed up the Wasserstein distance evaluation have been proposed, for instance via approximations relying on Sinkhorn regularisation [36], which can reduce the computational burden to *near-linear time*, while preserving accuracy [4]. In particular, the Sinkhorn method solves the following entropic regularisation problem,

$$P^{\gamma} = \underset{P \in \Gamma(X,X')}{\arg\min} \langle P, M \rangle - \gamma h(P). \tag{5.22}$$

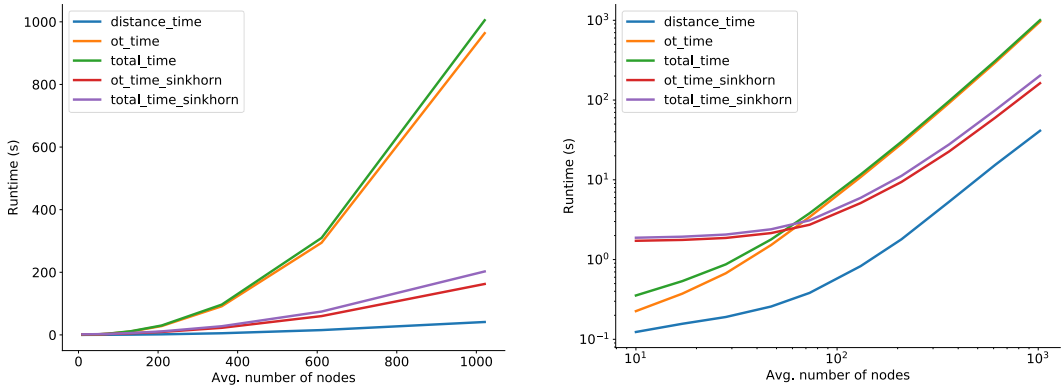
Such speedup tricks are incredibly useful in high-dimensional regimes, due to the regularization term making the optimization faster. However, the difference might be negligible in small size scenarios, if not harmful, due to the extra parametrization of the model.

In practice, looking at the runtime of WWL we observe that in our data sets the kernel matrix can be computed in a median time of 40 seconds, and since this is a one time operation, the theoretical computational burden is certainly not unbearable. Empirically, we see that for the continuous attributes our approach has a runtime comparable to GH. However, we expect the gap to grow for larger graphs, since GH was shown to empirically scale quadratically with the number of nodes [50]. Both HGK variants are considerably slower than WWL, as a consequence of the multiple hashing and stochastic component, requiring a certain number of iterations and repetitions for the method to converge.

To estimate the benefit of using approximations for the Wasserstein distance computation, we simulated a fixed number of graphs with a varying average number of nodes per graph. In particular, we generate random node embeddings for 100 graphs and varying the average number of nodes; for each graph, the number of nodes is taken from a normal distribution centered around the average. Then, we compute the kernel matrix on each set of graphs and compare the runtime of regular Wasserstein with the Sinkhorn regularised optimisation. As shown in Figure 5.6, the speedup can only be appreciated when the number of nodes grows to 200 or more, which is larger than the size of our data sets. The right plot in Figure 5.6 depicts a logarithmic scale, where it is clear that in small graphs regime running the Sinkhorn is more harmful than beneficial, given the hyperparametrization of the problem.

We perform a final experiment to evaluate the accuracy on one of the data sets (ENZYMES), with Sinkhorn versus regular Wasserstein. With the Sinkhorn approximation, we need to account for the extra γ parameter (Equation 5.22) to select via cross validation, that we choose in the range $\gamma \in \{0.01, 0.05, 0.1, 0.2, 0.5, 1, 10\}$. We observe that γ values selected most of the time are 0.3, 0.5, and 1; the accuracy on ENZYMES is 72.08 ± 0.93, which is slightly lower than regular Wasserstein but still above the state–of–the–art.

Figure 5.6: Runtime performance of the WWL Kernel computation step with a fixed number of graphs. We also report the time taken to compute the ground distance matrix as `distance_time`. Here, `total_time` is the sum of the time to compute the ground distance and the time taken to solve the optimal transport (ot) problem for the regular solver or the Sinkhorn-regularised one. The standard (left) and logarithmic (right) scales are shown.



(a) Standard scale

(b) Logarithmic scale

5.5 Discussion

In this chapter we presented a new family of graph kernels, the Wasserstein Weisfeiler–Lehman (WWL) graph kernels. Our method combines elements from optimal transport theory with an efficient WL propagation scheme, to obtain an informative and versatile similarity measure between node embeddings. We proved that WWL is PSD on categorically labelled graphs and we discussed the positive definiteness in the continuous scenario. We performed several experiments on graph classification settings and showed that WWL outperforms the state–of–the–art in the scenario with continuously attributed graphs, while is at least as performing as other methods in the categorical node label regime. We also evaluated the benefits of using approximations of the Wasserstein distance in terms of runtime and preservation of the predictive performance. Further investigating this aspect, would lead to natural extensions of our work to the large graphs regime, and is certainly an exciting future direction. Still with the aim of improving runtime, one should also think about deriving the explicit feature representation in the RKKS, as this would also provide a consistent speedup. On the theoretical side, major contributions could be made by defining bounds and conditions to ensure the positive definiteness of the WWL kernel in the case of continuous node attributes. Finally, neuralizaiton of the current method for the development of a new graph neural network, would be a promising and high–impact extension.

6 Adversarial graph neural networks

In the previous chapter we presented our novel Weisfeiler–Lehman Graph Kernel (WWL), that combined optimal transport theory with an iterative propagation scheme, to obtain improved node representations and boost the state–of–the–art in graph classification. We already observed that graph neural networks (GNN; Section 2.3) are a powerful alternative to graph kernels. The machine learning community recently witnessed an enormous explosion of these methods, in parallel with increasing abundance of large data sets. Nonetheless, graph kernels are generally limited to a small sample size regime, given the computational complexity of both the kernel generation step and classification learning algorithm (e.g. SVM). As argued in Section 5.4.4, for the WWL and for many other graph kernels, this bottleneck is negligible when the size and edge density of the graph is limited, while the whole pipeline from kernel generation to prediction can still be computed in a reasonable time. On the contrary, GNNs thanks to the backpropagation step and modern power resources (Graphical Processing Unit; GPU) have a considerably lower runtime, with the advantage of learning a complex function on a wide parameter space. However, as most neural network based approaches, GNNs tend to overfit on small data sets. Besides, for many relevant applications, and especially in medical and biologically related domains, it is often difficult to find and collect a large number of samples for a given task. In the field of chemoinformatics, this lack of data collection is particularly pronounced, due to the wide variety of existing compounds and the high cost and effort to evaluate their molecular properties or reactions with respect to the same prediction task. Despite generic graph molecular database exist [38, 68], only a subset of them in consistently annotated across tasks, while the majority is unlabelled, thus unsuitable, to be directly used in classification or regression problems.

Our goal to leverage all these aspects of the problem by introducing a domain adaptation scheme for graph neural networks, to simultaneously benefit from the high learning capabilities of GNNs and overcoming the small sample size issues. On one hand, transfer learning techniques aim to get better models on limited (target) domains, when the sample size is too small or prediction labels are missing, by learning information on a similar source domain and adapt it to the target [139]. These approaches have been largely developed in recent year, especially within deep learning applications, where their broad potential success is restricted by data scarcity [106, 201]. On the other hand, adversarial based networks were firstly introduced for generative models, to create new instances from a sample, by minimising an objective distinguishing between the gen-

erated and true data [73]. To put this all together, domain adversarial networks learn shared feature representations transferable across domains, with the adversary component aiming at optimally discriminate between samples coming from source and target data. We combine all these ingredients and present a novel framework to perform adversarial domain adaptation on graphs, the Adversarial Graph Neural Networks. We consider several applications on molecular data sets, from transfer to multi-task learning scenarios.

The remainder of this chapter is organised as follows. In Section 6.1 we introduce the transfer learning framework. In Section 6.2 we extend the various transfer learning scenarios to our domain of interest, Graph Neural Networks. The empirical evaluation and findings, including applications on different settings from molecular graph property prediction are presented in Section 6.3. We conclude with a critical discussion and ideas for future work in Section 6.4.

6.1 Transfer learning

We will now formalize the transfer learning framework and characterize multiple setting of it. Ee particularly focus on domain adaptation methods and their integration on neural networks via adversarial training. This section is partially adapted from Pan and Yang [139] and Ganin *et al.* [58].

Most of the machine learning methods are established on the underlying assumption that samples are drawn from the same distribution, accordingly to the data collection and modelling strategy. However, this is often not the case in practical applications where data are assembled at different iterations, individual samples often miss relevant information (e.g. class labels), background noise is not filtered or unknown, or the acquisition process is so complex to get a large cohort. For example, these are all common issues in MRI data, as we saw in Chapters 3 and 4, where sample size is limited and intrinsic differences between scanner and acquisition protocols make the modelling and comparison across samples challenging. The idea of *transfer learning* is to actually *transfer* the information across domains, generally from a *source* to a *target*, and exploit the knowledge learned from a different problem to solve the new one. The source and target domain belong to different but related distributions and in the classical scenario more knowledge is available on the source, while limited information is given on the target. The term transfer learning is quite broad and it has been given different names and/or subfields such as inductive transfer learning, knowledge transfer, domain adaptation, multi-task learning. The latest one is a closely related area where the goal is to learn a join model across multiple tasks. Sometimes, multi-task learning is presented as an instance of transfer learning. For the purpose of this thesis, we will characterize different transfer learning scenarios, such as domain adaptation, multi-task learning and supervised transfer learning. We begin by providing a general introduction to the topic and we will later present the different settings of transfer learning which are of interest for our method.

Let us assume we are given a source domain $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{P}(X_s))$ on an input feature space $\mathcal{X}_s \sim \mathcal{P}(X_s)$ with probability distribution $\mathcal{P}$, samples $X_s = \{x_{s,1}, \dots, x_{s,m}\} \in \mathcal{X}$ and labels $Y_s = \{y_{s,1}, \dots, y_{s,m}\} \in \mathcal{Y}_s$. A model on the source can be defined as

$$\begin{aligned} f &: \mathcal{X}_s \mapsto \mathcal{Y}_s \\ f(x_{s_i}) &= \hat{y}_{s_i}, \end{aligned} \tag{6.1}$$

with associated *task* denoted as a pair $\mathcal{T}_s = \{Y_s, f(\cdot)\}$. The function f is then *adapted* or *transferred* to a function $\tilde{f}$ and applied on a target domain $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{P}(X_t))$, to predict the target labels:

$$\tilde{f}(x_{t_j}) = \hat{y}_{t_j}, \tag{6.2}$$

with $Y_t = \{y_{t,1}, \dots, y_{t,n}\} \in \mathcal{Y}_t$. With this notation in mind, the transfer learning problem can be summarised in a definition.

Definition 6.1 (Transfer learning [139])**.** For a source domain $\mathcal{D}_s$ and task $\mathcal{T}_s$, the aim of transfer learning is to improve the predictive model $\tilde{f}(\cdot)$ on a target domain $\mathcal{D}_t$ with learning task T_t, exploiting the knowledge from $\mathcal{D}_s$ and $\mathcal{T}_s$, where $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$.

A characterization of the source and target domain, in terms of their distributions and label availability provides a characterization of the different settings of transfer learning.

6.1.1 Supervised transfer learning

The supervised transfer learning, or inductive transfer learning, is one of the most common scenario: given a source and target domain, both labelled, we learn a model on the source and adapt it to the target. Typically, the distribution of the input data and/or labels need to be aligned for the transferring to be successful. Specifically, we consider the setting where all $\mathcal{T}_s$, $\mathcal{T}_t$, $\mathcal{D}_s$, $\mathcal{D}_t$ are given, $\mathcal{T}_s \neq \mathcal{T}_t$ while $\mathcal{D}_s$ and $\mathcal{D}_t$ can be either the same or not. In our application, the feature space of source and target is the same, i.e. $\mathcal{X}_s = \mathcal{X}_t$, but the distribution of data itself could differ, $\mathcal{P}(X_s) \neq \mathcal{P}(X_t)$. The model functions f, $\tilde{f}$ can take different formats: in deep learning this is a neural network, in classical methods it could be a supervised classification or regression model, e.g. SVM or linear regression.

Pre-training in deep learning. In the context of deep learning, the supervised pre-training approach can be seen as an instance of the supervised transfer learning, though for our purpose we will characterize it as a separate application (see Section 6.2.4). The parametrized network model read as

$$\hat{y}_s = f(\theta, X_s), \tag{6.3}$$

for parameter space θ. After training, the optimal parameters are learned $\hat{\theta}$ and used as input for the transferred function $\tilde{f}$

$$\hat{y}_t = \tilde{f}(\hat{\theta}, X_t), \tag{6.4}$$

i.e. $\tilde{f}$ has input parameters $\hat{\theta}$ initialized with a pre-training on the source. It is possible to either fix θ or perform a further training on the target. A standard approach is to fine-tune only the last layers and freeze the initial ones, from the rationale that initial layers represent shared features representative of the input data while the last layers output target specific features. This setting is commonly used in deep learning applications, when a lot of labelled data is available in the source and limited samples are provided in the target.

6.1.2 Multi-task learning

The multi-task learning scenario is similar to the supervised transfer learning, with the difference that all the the tasks are learned simultaneously while the transfer learning focuses to improve performance on the target task of interest. Suppose we are given T different tasks and each task $\mathcal{T}_j = \{Y_j, f(\cdot)\}$ is defined on a domain $\mathcal{D}_j = (\mathcal{X}, \mathcal{P}(X))$, where $\mathcal{D}_0 = \mathcal{D}_1 = \dots \mathcal{D}_T$, i.e. same domain across task. In the general formulation, the function f can have a *shared* and *task specific* component and is modelled simultaneously on the multi-task data, with the goal to improve performance over the single task specific model. The general multi-task learning function can be written as

$$\hat{y}_j = f(X) = f_{sh}(X) + f_j(X), \text{ for all } j = 1, \dots, T \tag{6.5}$$

where f is decomposed into f_{sh} and f_j, the shared and task-specific components, respectively. Modifications of the model are possible, for instance one might restrict the shared component to a subset of the tasks, given by expert domain knowledge and relatedness across tasks.

6.1.3 The scenario of domain adaptation

We now consider a different scenario, which we identify as the *domain adaptation*. Suppose we have related source and target domain with the same input feature space; we are also given target tasks $\mathcal{T}_t$ and labels Y_t, but the source labels are not available. We note that Pan and Yang [139] define this setting as an instance of inductive transfer learning, or self-taught learning. We call this the *domain adaptation* or *unsupervised domain adaptation* scenario, given that no label is available on the source and no assumption can be made on the similarity of the tasks. Here, the model should reflect that no information can be exploited from the source domain, but we can exploit both source and target features in the predictive model:

$$\hat{y}_t = \tilde{f}(X_s, X_t). \tag{6.6}$$

6.1.4 Domain adversarial training

We conclude this section by discussing an extension of the classical transfer learning framework, specifically developed for deep learning models and exploiting recent advances in adversarial networks. The numerous attempts made to combine transfer and multi-task learning methods with adversarial based approaches, aim to encourage the learning of shared and task specific features by including a domain classifier to separate them [58, 111, 175]. Adversarial networks have been first introduced in the context of generative models (GANs; [73]), integrating the generator with a *discriminator* layer. While the generator creates sample data, the discriminator is a classifier whose prediction task is to distinguish if the input data is fake or real. On a related scope, adversarial layers for transfer learning aim to separate instances of the source and target domain, simultaneously wishing for the learned feature representations to be *shared* between source and target, such that the source model f can be easily adapted to $\tilde{f}$ on the target. In the original model proposed by Ganin *et al.* [58], the discriminator makes use of a *gradient reversal layer*, changing the sign of the gradient during backpropagation and acting as an identity in the forward pass. The ultimate loss is a weighted *subtraction* of the task classification loss and discrimination (domain) loss, i.e. when the label corresponds to source and target. Practically, the loss function of the domain classifier $\mathcal{L}_{dom}$ is multiplied by a weighting constant λ and added to the classifier loss $\mathcal{L}_{cls}$, while a negative update is performed during backpropagation:

$$\left(\mathcal{L}_{dom}; \frac{\partial \mathcal{L}_{dom}}{\partial \theta_{dom}}\right) \implies \left(-\lambda \mathcal{L}_{dom}; -\lambda \frac{\partial \mathcal{L}_{dom}}{\partial \theta_{dom}}\right) \tag{6.7}$$

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{dom} \tag{6.8}$$

6.2 Adversarial layers for graph neural networks

We will now leverage the concepts described in Section 6.1 and develop them into a novel framework, that combines graph neural networks with an adversarial based transfer learning approach. We present different instances of our method, corresponding to the three transfer learning setting introduced: supervised transfer learning, unsupervised domain adaptation, and multi-task learning.

We recall that our problem of interest is a graph–level prediction task: given a set of graphs with associated label, we aim to infer their structural and task specific similarities to optimally model the prediction. In the context of transfer learning this is the target data set, where we want to improve the performance. More formally, let $\mathcal{G}_t = \{G_1, \ldots, G_n\}$ be a set of graphs in the *target* domain, with labels $Y = \{y_1, \ldots, y_n\} \in \mathcal{Y}_t$. In the standard machine learning scenario, we model the prediction problem by learning a function $f(G) = Y$. We consider this to be a *challenging* target prediction, due to lack of information or limited number of data points. To get back to the chemoinformatics example, where the graphs represent small molecules, it is a common issue

that only a subset of the data has been screened for particular molecular tasks, leading to limited labelled data. However, the overall availability of chemical compounds is huge, but the variety of different annotation (task) is also large, leading to an inhomogeneous big molecular cohort. This is indeed a perfect setting for transfer learning, where information from different or inexistent labels need to be exploited, in order to learn a better model on the small, complete, target data. We denote the *source* domain $\mathcal{G}_s = \{G_1, \ldots, G_m\}$, with $m > n$ (ideally $m >> n$), with labels $Y = \{y_1, \ldots, y_m\} \in \mathcal{Y}_s$. We will consider different scenario, depending on the format of Y: (1) the source label come from a different task than target; (2) the source labels are not collected (unsupervised). In our application, the two domains are clearly related, representing small molecules modelled as graphs. Furthermore, with consistent input definition, the feature space of source and target coincides. We summarize the main idea of our method, as learning a GNN that takes as input *source* and *target* molecules, and learns a *global* graph molecular feature representation to employ in the target classification layer. The speculation is that this graph feature simultaneously incorporates a *shared* and task specific component, and is an improvement over the representation that would be obtained if the target graphs (only) were given as input to the GNN. The gradient reversal layer is included in the model, to encourage the embeddings to be as agnostic as possible of the input domain, while representative of the target task.

6.2.1 Adversarial graph neural networks

Consider the union of *source* and *target* domain, denoted as $\mathcal{G}_{s,t} = \mathcal{G}_s \cup \mathcal{G}_t$, and let $X_{G_{s,t}} \in \mathbb{R}^{n_G \times p}$ be the input feature of graph G with n_G being the number of nodes in graph G and p the feature dimension. As a first step, we define a generic GNN model, which we interpret as the feature extractor GNN:

$$\begin{aligned} &\text{GNNext}(\cdot; \theta_\phi) : \mathbb{R}^{(n_G \times p)} \mapsto \mathbb{R}^{p_o} \\ &\text{GNNext}(G_i; \theta_\phi) = \phi(G_i), \text{ for each } G_i, \in \mathcal{G}_{s,t}. \end{aligned} \tag{6.9}$$

with parameters θ_ϕ, and output graph features $\phi(G) \in \mathbb{R}^{p_o}$. The next step of a GNN is to use the extracted features as input for the classification layer, i.e.

$$\text{GNNcls}(G_i; \theta_f) = f(\text{GNNext}(G_i)) = f(\phi(G_i)) = \hat{y}_i \tag{6.10}$$

where f is a general classification function that can include multiple linear and non-linear layers with corresponding parameters θ_f. The feature extractor and classification part are then trained together in an end-to-end fashion, with the following loss function:

$$\mathcal{L}_{cls}(\mathcal{G}_s, \mathcal{G}_t, \mathcal{Y}_t) = \mathcal{L}_{cls}(f(\phi(\mathcal{G}_t)), \mathcal{Y}_t). \tag{6.11}$$

Here, we are giving as input either a source or target graph, however the loss function is only defined on the target data. Therefore, in this formulation the source data is not contributing to the learning step, but in practice is used as *phantom* data, i.e. being ignored by the model. To benefit from the extra information provided by the source, we

need to define a loss that also incorporate a learning task on the source. This is achieved with the adversarial layer (GNNadv), taking as input the graph feature representations obtained with GNNext and constructing a new classification layer(s), whose prediction task is to discriminate between a source and target instance. GNNadv uses a *gradient reversal layer* [58], meaning that in the backpropagation step the sign of the gradient is flipped, to enforce the graph representation obtained via GNNext to be as general as possible and indistinguishable between source and target, that is $\phi(\mathcal{G}_s) \sim \phi(\mathcal{G}_t)$. Suppose the domain label is given by $\mathcal{Z}_d = \{z_1, \ldots, z_m, \ldots, z_{m+n}\}$, where $z_i = 0$ if $G_i \in \mathcal{G}_s$ and $z_i = 1$ if $G_i \in \mathcal{G}_t$. Then, we define $\text{GNNadv}(\cdot, \theta_d)$ with parameters θ_d as:

$$\text{GNNadv}(G_j; \theta_d) = d(\text{GNNext}(G_j)) = d(\phi(G_j)) = \hat{z}_j \tag{6.12}$$

where d is the general classification function, encoding linear or non-linear layers. The domain adversarial component of the loss is defined on both *source* and *target* instances as follows:

$$\mathcal{L}_{dom}(\mathcal{G}_s, \mathcal{G}_t, \mathcal{Z}) = \mathcal{L}_{dom}(d(\phi(\mathcal{G}_{s,t})), \mathcal{Z}). \tag{6.13}$$

Ultimately, we combine the domain classifier and task specific classifier to postulate the optimisation problem of GNNadv, as a joint minimisation of the two components:

$$\begin{aligned} \min_{\theta_\phi, \theta_f, \theta_d} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{cls}(f(\phi(G_i; \theta_\phi); \theta_f); y_i) \\ -\lambda \sum_{j=1}^{n+m} \mathcal{L}_{dom}(d(\phi(G_j; \theta_d); \theta_f); z_j) \end{aligned} \tag{6.14}$$

where λ is a weighting constant and $G_i \in \mathcal{G}_t$, $y_i \in \mathcal{Y}_t$, $G_j \in \mathcal{G}_{s,t}$, $z_j \in \mathcal{Z}$. We can rewrite it more compactly to define the loss of GNN-ADV as:

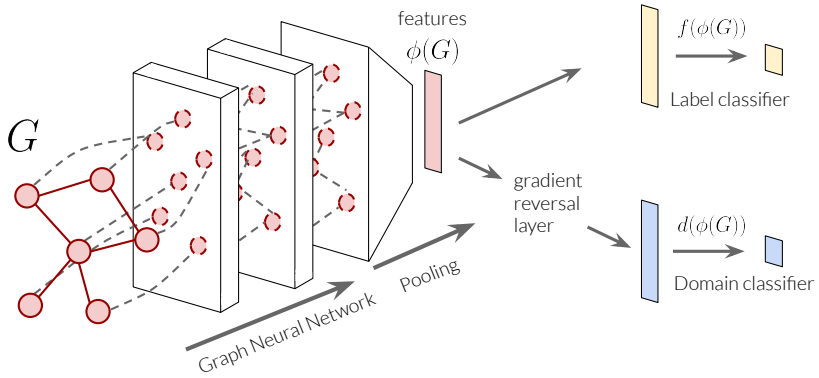
$$\mathcal{L}_{adv}(\mathcal{G}_s, \mathcal{G}_t, \mathcal{Y}_t, \mathcal{Z}) = \mathcal{L}_{cls}(\mathcal{G}_t, \mathcal{Y}_t) - \lambda \mathcal{L}_{dom}(\mathcal{G}_s, \mathcal{G}_t, \mathcal{Z}). \tag{6.15}$$

An overview of the GNN-ADV architecture is depicted in Figure 6.1. The GNN-ADV architecture is agnostic of the source class labels, therefore it is a suitable solution for the unsupervised domain adaptation scenario. We will discuss in Section 6.2.2 how to incorporate labels in the source domain to cover the supervised transfer learning setting.

TASK-BASED VERSUS SHARED FEATURES

Our architecture implicitly assumes the existence of a universal graph molecular feature $\phi(G)$, incorporating hidden graph properties and patterns of the source and target domain, while being representative of the target task of interest. While this could be achieve in specific problems, it an oversimplification in many real–world applications. Indeed, it is hard to learn features that are simultaneously general enough and task specific. In the classical neural networks this issue is addressed by combining the information on multiple layers or concatenating intermediate features. For example, in

Figure 6.1: Schematic view of adversarial Graph Neural Networks (GNN-ADV).



a multi-layer deep architecture for image analysis, it is known that the initial features encode general structural properties of the data, while the deeper layers generate task specific embeddings. Therefore, it might be more convenient to separate the feature extractor step after a fixed number of common layers to obtain separated task specific and shared features, which are then properly combined in the classification layers. Following this rationale, we propose an alternative architecture that generates for each graph two set of features, $\phi(G)_{Ts}$ and $\phi(G)_{Sh}$; the initial layers of the network are shared and then at a deeper level they separate, to obtain task specific and shared representations. This is achieved by separating the feature extractor GNN as:

$$\text{GNNext}_{Ts}(G_i; \theta_{\phi_{Ts}}) = \phi(G_i)_{Ts} \tag{6.16}$$

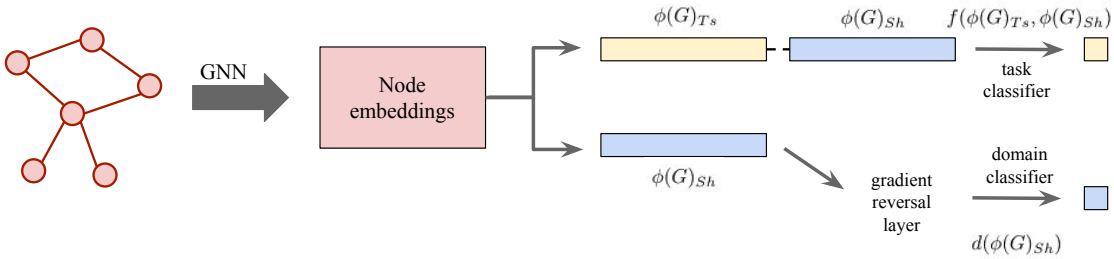
$$\text{GNNext}_{Sh}(G_i; \theta_{\phi_{Sh}}) = \phi(G_i)_{Sh}, \tag{6.17}$$

with $G_i \in \mathcal{G}_{s,t}$. We denote by GNNext_{Ts} the task specific layers and by GNNext_{Sh} the extractor for the shared representation. On one hand, the shared features $\phi(G)_{Sh}$ are used as input for the adversarial layer and optimised via the domain classification loss. On the other hand, a concatenation of $\phi(G)_{Ts}$ and $\phi(G)_{Sh}$ is used in the classification part of the GNN to perform the target prediction task. In practice, the layers of the GNN extractors, and therefore their parameters, are joint at the beginning and separate at a deeper level. This is integrated in our architecture which we denote as GNN-ADV-TS, as shown Figure 6.2.

6.2.2 Supervised transfer learning

Until now, we did not included any task related label information on the source domain and only considered the unsupervised scenario. In the field of chemoinformatics, the same compound may be screened for multiple properties (multi-task learning) or different molecules are available for varying prediction tasks. Then, we can exploit different functional labels collected on the source data to enhance the model prediction on the target. However, the issue still remain that the labels need to be related in order

Figure 6.2: Schematic view of task-shared adversarial GNN.



to get a successful transfer. Using unrelated labels might result in task specific embeddings which are not suitable to be shared across domains, hereby increasing the level of noise in the model. We will further investigate this aspects in our experimental evaluation and discussion later throughout this chapter. For the theoretical development, we assume that the task are similar enough to allow for a supervised transfer learning approach. We then refine our architectures to also take the new source label into account. This is achieved with a simple adjustment in Equation 6.14, including graphs from both source and target in the classification loss, therefore rewriting the optimization problem as:

$$\mathcal{L}_{cls}(\mathcal{G}_s, \mathcal{G}_t, \mathcal{Y}_t, \mathcal{Y}_s) = \mathcal{L}_{cls}(f(\phi(\mathcal{G}_{s,t})), \mathcal{Y}_{s,t}). \tag{6.18}$$

6.2.3 Multi-task adversarial learning

We now consider the multi-task learning scenario as a straightforward extension of our framework, which is particularly relevant if different functional tasks are available from the same domain. In this setting, is especially important to account for the difference between shared and task based representation: it would be over optimistic to look for a graph representation that is shared across input structures, and simultaneously relevant for multiple tasks of interest. Following a similar rationale as before, we consider that the early layers capture general graph molecular properties, while the deeper layers create a feature representation that is associated to the task. Therefore, our multi-task learning model expands over the task-shared architecture (GNN-ADV-TS), such that the weights of the network are shared in the first layers and at a deeper level generate different sets of graph features. We then obtain a shared representation and multiple task specific embeddings. For each task, the adversarial layer distinguishes between

shared and task related features, to then average their contribution. Mathematically, we express the loss as:

$$\begin{aligned}\min_{\theta_{\phi_1,\dots,\phi_T},\theta_{f_d},f_1,\dots,f_T,\theta_d} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{T}\sum_{j=1}^{T}\mathcal{L}_{cls}(f(\phi(G_i;\theta_{\phi_j});\theta_{f_j});y_{i,j}) \\ -\lambda\sum_{i=1}^{n}\frac{1}{T}\sum_{j=1}^{T}\mathcal{L}_{dom}(d(\phi(G_i;\theta_d);\phi(G_i;\theta_{\phi_j});\theta_{f_d});z_i).\end{aligned} \tag{6.19}$$

The first part of Equation 6.19 is an average of the classification loss evaluated independently on each task, with respect to their specific feature $\phi(G_i)_{T_j}$, for each $j = 1,\dots,T$ with T being the number of tasks, while $y_{i,j}$ is the label of sample i per task j. The second part of the loss incorporates the discriminator, to distinguish between graph features $\phi(G_i)_{T_j}$ and shared features $\phi(G_i)_{Sh}$, where z_i is the corresponding domain label encoding the type of feature. The individual contributions are averaged across tasks. In practice, as for the GNN-ADV-TS model, we will use a concatenation of the shared and task-specific features as input for the classification layer. We call this the graph neural network adversarial multi-task architecture (GNN-ADV-MT). Additionally, we will also consider the vanilla variant, the multi-task graph neural network (GNN-MT). This is obtained by disregarding the second component of Equation 6.19, thus only considering the label classification loss.

6.2.4 Pre-training graph neural networks

In deep learning, one of the most simple transfer learning strategies is a pre-training of the neural network on the source domain with fine-tuning on the target [201]. While complex variants exist, the vanilla approach to pre-training requires little extra technical development from the original architecture. The idea is to use the same model on the source and target; when training on the target, the hyperparameters are initialized with the optimal weights obtained on the source. Usually, the initial layers are freezeed (i.e. no training learning on the target), while the deeper layers are optimized for some additional iteration on the target domain. Other common tricks, for example regularizing the loss or sharing sub–parts of the network, can also be employed depending on the problem of interest and domain knowledge [113, 114]. This approach comes with several advantages. First, the pre–training only needs to be performed once, hence getting a speed up of training time on the target data which will require fewer epochs to learn and fit the model. Second, pre-training on a large dataset can prevent overfitting in small sample size regimes, where deep learning architectures often present limitations.

A recent study by Hu *et al.* [89] proposed several strategies to pre–train GNNs, showing that a successful approach requires pre–training to be performed jointly at graph and node-level tasks. They also reported that straightforward pre–training approaches, treating node and graph-level representations separately, can prevent generalisability of the learned model and lead to negative transfer [89, 152]. The node–level pre-training

Table 6.1: Complete information about the target data sets.

Data set	Category	# Compounds	# Tasks	# Classes
BACE	Biophysics	1513	1	2
BBBP	Physiology	2039	1	2
CLINTOX	Physiology	1478	1	2
SIDER	Physiology	1427	27	2
TOX21	Physiology	7831	12	2

is a self-supervised approach, ensuring that nodes with similar structural characteristics are mapped to similar embedding representations. The graph–level supervised pre-training encourages to learn global graph features, to promote their transferability across tasks. We integrate these pre–training strategies within our method, creating additional variants of our approach. We define a model by pre–training the main GNN-Next on the source only, and successively fine–tune it jointly with the adversarial layer on the source and target. The variants of our approach are called: preGNN-ADV-G, preGNN-ADV-N, preGNN-ADV-N-G, depending on whether the pre-training step is applied at graph–level, node–level, or both, respectively. Hu *et al.* [89] proposed several node–level pre–training strategies, we choose to employ the ContextPrediction, as the authors reported it to be the most effective. For additional details on the methodology and architecture we refer the reader to the original publication [89].

6.3 EXPERIMENTS

In this section we evaluate the performance of GNN-ADV and its variants on multiple molecular data sets. We discuss the different scenario of transfer learning and empirically assess the impact of integrating the adversarial layer.

6.3.1 DATA SETS

TARGET DATASETS. As target, we consider 5 data sets with various size and molecular property prediction labels: BACE, BBBP, CLINTOX, SIDER, TOX21. BACE is from the biophysics domain, containing qualitative binding results for a set of inhibitors of human β - secretase 1 [166]. The other data sets all belong to the physiology category: BBBP has binary labels of blood-brain permeability [120]; CLINTOX encodes qualitative data of drugs approved by the FDA and those that have failed clinical trials for toxicity reasons [133]; SIDER is a database of marketed drugs and adverse drug reactions (ADR), grouped into 27 system organ classes [107]; TOX21 reports toxicity measurements on 12 biological targets, including nuclear receptors and stress response pathways [31]. In terms of the prediction task, all the data are designed for the graph–level classification problem, either single- or multi-task, as detailed in Table 6.1. The data can

be found in the MoleculeNet database [197] and we extracted them from the DeepChem repository [147].

SOURCE DATASETS. Different source data sets are used in the unsupervised and supervised settings. For domain adaptation we used a curated version of the ChEMBL [38, 68] database containing $\sim$ 1.8 million compounds. For supervised transfer learning we create source data sets by mixing staamples of different, but related, functional tasks from the targets. In particular, BBBP, CLINTOX, SIDER, and TOX21 all contain labels from physiology measurements, such as toxicity information or drug reaction. Therefore, when predicting on one target data set we can create combined source from the other ones.

DETAILS ON DATA SPLITTING, PROCESSING AND FEATURES

For all data sets we exclude graphs having more than one connected components, as well as singletons, i.e. graphs consisting of a single isolated node. On the source CHEMBL we further exclude small graphs with less than 10 nodes, to match the distribution of the target data and to ensure detection of meaningful and informative patterns. For the node features, we use a similar input as Duvenaud *et al.* [45] and concatenate one-hot-encoding arrays, resulting in 75 features of different atom properties obtained from RDKit[1] [109]: atom type, degree, implicit valence, hybridization type, number of radical electrons, aromaticity, number of attached hydrogen atoms, formal charge. To guarantee generalisation ability of our algorithm on out–of–distribution samples, we use a *scaffold* train, validation and test splitting [147]. With a *scaffold* split, molecules are divided according to their substructure, such that structurally dissimilar molecules are placed in different splits, making the prediction task on the test and validation set more challenging than a random split.

SELECTING DATA FOR PRE-TRAINING. For the pre-training step, we randomly sample two subsets of 500k molecules from CHEMBL, independently for the node-level and graph-level strategy, respectively. These molecules are then excluded as candidate source data for GNN-ADV, to avoid the pre-training and adversarial components of the network to overfit on the same samples. For the supervised graph-level learning task we use a set of 111 molecular predictors extracted from RDKit.

6.3.2 EXPERIMENTAL SETUP

In all our experiments we use a 5-layer GIN (see Section 2.3.2) model with parameter $\epsilon = 0$ and 300 hidden units as the main feature extractor GNNext [198]. In the multitask setting (GNN-MT; GNN-ADV-MT) and in the GNN-ADV-TS, the first 4 layers are shared and the last layer is split to generate task specific and shared features. The GNNcls layer is a linear classifier with a *softmax* activation function, where we use the

[1] https://www.rdkit.org

negative log likelihood as a loss function. The GNNadv network consists of 2 fully connected linear layers, and the prediction is performed by a *sigmoid* activation with cross entropy loss function. As a baseline, we use the vanilla GNN trained and evaluated on the target data only; this is also a GIN model ad the parameters are chosen as in the adversarial counterpart. In the pre–training approaches, we follow the recommendation in Hu *et al.* [89] and use a 3-layer GIN for the contextGNN with inner and outer radius being set to $r_1 = 1$ and $r_2 = 4$, respectively; in this step, we further exclude small graphs for which the context was not computable for these parameters. All the baselines and the GNN-ADV models are trained for 100 epochs, while the pre–training runs for 50 epochs, using the Adam optimiser, a learning rate of 0.001, and a batch size of 32. Accordingly to the methodology described in Section 6.2, we define 4 different experimental settings.

1. **Unsupervised domain adaptation (Scenario 1).** We apply GNN-ADV on a joint dataset consisting of unlabelled source (CHEMBL) and labelled target, with various functional molecular prediction tasks.

2. **Supervised transfer learning (Scenario 2).** In this setting GNN-ADV is trained as for Scenario 1, with the inclusion of functional labels on the source data. We investigate the performance on the three small sample size regime data sets from the physiology target domain (BBBP, CLINTOX and SIDER). For each target data, the remaining two plus TOX21 are used as source, resulting in the following transfer learning tasks: BCT $\rightarrow$ SIDER; BST $\rightarrow$ CLINTOX; CST $\rightarrow$ BBBP (here the acronyms reflect the data used as source, e.g. BCT = BBBP + CLINTOX + TOX21).

3. **Multi-task learning (Scenario 3).** Here we evaluate the multi-task adversarial GNN on TOX21 and SIDER, the two datasets from our cohort for which prediction labels from multiple tasks available.

4. **Pre-training (Scenario 4)**. The pre-training strategy is applied on the vanilla GNN and as a variant of GNN-ADV (Scenario 1). In this experiment, the GNN graph and node level is pre-trained on CHEMBL and the learned model is used to initialise the target GNN [89].

EVALUATION. The batches are balanced to guarantee an equal proportion of source and target samples. For training, we randomly sample from the source domain the same number of compounds available in the target split. In the unsupervised domain adaptation setting (Scenario 1) the batches are balanced with respect to the target class label and source, with a proportion of $1/3$ each, and only the samples from the negative class are passed through the adversarial. In the supervised transfer learning (Scenario 2), we used standard balanced batches, with half of the sample coming from the target, and source, respectively. The parameter λ in GNN-ADV is defined according to the guidelines in [58], thus updated at each epoch during training, with the following rule:

$$q = \frac{epoch}{NumEpoch}; \ \lambda = \frac{2}{1 + \exp(-10 \cdot q)} + 1, \tag{6.20}$$

Table 6.2: Classification results in the unsupervised domain adaptation scenario. Test AUROC performance of our method with the two architectures: GNN-ADV and GNN-ADV-TS. The winning approach is highlighted in bold and an * indicates a statistically significant difference to the baseline.

Dataset	CLINTOX	SIDER	BACE	BBBP	TOX21
GNN	80.17 ± 3.42	53.76 ± 4.08	70.20 ± 3.38	$\mathbf{62.95 \pm 2.61}$	61.30 ± 4.13
GNN-ADV	$\mathbf{80.70 \pm 5.80}$	$\mathbf{58.78 \pm 4.75}^*$	70.67 ± 2.73	62.24 ± 2.14	63.85 ± 3.97
GNN-ADV-TS	70.49 ± 5.48	55.46 ± 6.40	$\mathbf{75.55 \pm 2.12}^*$	61.50 ± 2.75	$\mathbf{66.04 \pm 3.81}^*$

where *epoch* is the current epoch at training time and *NumEpoch* is the total number of epochs. We evaluate the performances on validation set using the best average area under the ROC curve (AUROC) over 10 random initialisation of the model; we then report the corresponding performance on the test set. For the multi-task datasets (TOX21; SIDER), we average the test performance over tasks (Scenario 1 and 2); in Scenario 3, this is not necessary, since the average performance is already included in the loss function (Equation 6.5).

6.3.3 RESULTS ON MOLECULAR DATASETS

UNSUPERVISED DOMAIN ADAPTATION

We compare the two instances of our approach GNN-ADV and GNN-ADV-TS with the GNN vanilla baseline and report our main findings in Table 6.2. Overall, we observe that adding an adversarial layer is beneficial in 4 out of 5 datasets. The superiority is mostly pronounced in SIDER, BACE, and TOX21, where GNN-ADV-TS improves by 2%, 5%, and 5%, respectively. However, the best performance on SIDER is achieved with GNN-ADV, with also a +%5 gap over the vanilla baseline. On CLINTOX, GNN and GNN-ADV are on par, while the task-shared representation harms the performance by 10%. We do not have a conclusive reason yet to justify this behaviour, but we will also observe oscillating performance on this data set in the supervised transfer learning scenario. We speculate that this could be due to the peculiarity of the data distribution of the functional task, which represent a qualitative measure rather than an effective physiological property, hence not aligning with the source domain. On the BBBP data, a vanilla GNN appears to be sufficient to achieve high classification performance.

SUPERVISED TRANSFER LEARNING

In the supervised setting, we exploit the functional tasks in the source domain, which therefore need to be related to the target labels, if we aim for the task specific features learned in the source to be transferable to the target. As previously described, this is achieved by restricting our analysis to the physiology data domain. As for the unsupervised setting, we compare GNN-ADV and GNN-ADV-TS with the vanilla baseline and report the average AUROC curve on the target data sets in Table 6.3. In BBBP and

Table 6.3: Classification results in the supervised transfer learning scenario. Test AUROC performance of our method with the two architectures: GNN-ADV and GNN-ADV-TS.

Dataset	BST → ClinTox	BCT → SIDER	CST → BBBP
GNN	**80.17 ± 3.42***	53.76 ± 4.08	62.95 ± 2.61
GNN-ADV	69.28 ± 6.40	**59.31 ± 7.39**	**63.59 ± 1.18**
GNN-ADV-TS	55.33 ± 6.09	56.09 ± 4.42	60.15 ± 2.79

SIDER, we observe that GNN-ADV outperforms GNN, despite the standard deviations overlap in both cases. This suggests that the embeddings are successfully transferred from source to target task. However, the benefit is not as pronounced as for the unsupervised scenario, possibly due to the source size being smaller or the tasks being too inhomogeneous. As for Scenario 1, the CLINTOX data set seems to be an outlier in this cohort with respect to the transferability of graph related features. Remarkably, the best overall results on SIDER and BBBP are achieved in this scenario by GNN-ADV. This is a key observation, implying that the relatedness of the functional tasks makes the explicit modelling of separated tasked based and shared features obsolete.

MULTI-TASK LEARNING

According to the availability of multiple functional tasks in our cohort, we can only evaluate the adversarial multi-task learning on two data sets. We first observe that the vanilla multi-task GNN-MT, with shared features at the early layers and task-specific embeddings at a deeper level, achieves an AUROC of 54.68 ± 0.62 and 60.49 ± 0.67 on SIDER and TOX21, respectively. This is very close to the vanilla GNN (Table 6.2) and in agreement with previous literature, which already suggested that on these molecular datasets multi-task deep architectures are not always beneficial [148]. However, when adding the adversarial layer, a slight improvement can be observed on TOX21 (AUROC = 62.41 ± 0.84), showing the potential of our method. Besides, GNN-ADV-TS and GNN-ADV both perform better, without the need to explicitly model the multi–task component. Nevertheless, it is worth to note that, as opposed to Scenario 1 and Scenario 2, the standard deviations are very small, indicating that our findings are more robust and reliable. This was also expected, since we are considering a single data set the input is less heterogeneous.

6.3.4 PRE-TRAINING ADVERSARIAL GNN

We conclude our empirical analysis by investigating the impact of pre-training graph neural networks, either with or without the adversarial layer. Results are reported in Table 6.4. Unsurprisingly, except for CLINTOX, the pre-training provides higher performance than the vanilla GNN. Overall, the adversarial layer has a mixed effect when paired with pre-training: we do not see a clear pattern of either improvement or drop in classification performance. Such effect could be due to the choice of the pre-training task, given that we used molecular predictors which are built-in properties of

Table 6.4: Classification results with pre-training. Test AUROC performance of our method with the pre-training variants at node-level (preGNN-ADV-N), graph-level (preGNN-ADV-G) and node-level+graph-level pre-training (preGNN-ADV-N-G). The baselines are denoted as GNN-N, GNN-G and GNN-N-G, with node-level, graph-level, and node-level + graph-level pre-training.

Dataset	clintox	sider	bace	bbbp	tox21
preGNN-G	74.64 ± 5.40	57.67 ± 4.79	72.39 ± 2.06	62.19 ± 1.04	66.08 ± 3.22
preGNN-ADV-G	72.76 ± 5.48	59.31 ± 2.47	74.49 ± 2.03	62.09 ± 1.07	65.74 ± 3.76
preGNN-N	75.54 ± 3.99	57.30 ± 4.43	70.81 ± 2.12	64.10 ± 2.98	62.31 ± 2.13
preGNN-ADV-N	74.19 ± 3.84	55.03 ± 5.53	73.71 ± 2.85	63.31 ± 3.21	63.40 ± 4.69
preGNN-N-G	73.96 ± 4.22	56.36 ± 6.41	74.24 ± 2.19	60.69 ± 1.14	68.01 ± 5.41
preGNN-ADV-N-G	69.41 ± 6.68	55.47 ± 4.11	75.14 ± 1.28	61.07 ± 2.15	62.32 ± 5.43

the molecules that are unrelated to the functional target task. We further observe that our experimental setting is slightly different from Hu *et al.* [89], indeed we find that the combination of node and graph level pre-training is not always beneficial.

6.4 Discussion

Runtime analysis. Our GNN-ADV method and its variants require little additional runtime compared to the vanilla GNN approach. The computational complexity equals the complexity of the main GNNext and GNNcls models plus domain layers, since no additional operations are performed. The higher runtime is only conditional on the size of the source domain in the training set. In our experiments, we employ a source sample size proportional to the target, with a $1:1$ ratio; since our datasets are already small, we observe that the computational bottleneck of adding the source domain is relatively limited. The pre-training is computationally expensive, especially when the graph-level and node-level is combined. However, as it is only performed once, the targte runtime itself is not impacted

Novelty and empirical findings. To the best of our knowledge, our work is the first one proposing adversarial layers on GNN to solve the supervised graph-level classification problem via transfer learning. We presented multiple variants of adversarial graph neural networks with applications in different settings, including multi-task learning, supervised and unsupervised domain adaptation. Overall, we observe that the proposed approach leads to interesting empirical considerations and can improve classification performance on multiple target data sets, when the sample size is limited or in multi-task learning scenarios. Nevertheless, negative transfer can also occur [152], in particular when the prediction task in the target data set is not well defined. We empirically demonstrated that our approach overcomes these limitations if the data from source and target are aligned, since the functional task of interest and the similarity in the molecular graph domain are crucial elements to successfully transfer information

across data sets. We further observe that most of the methods exhibit a rather large standard deviation. We speculate the reasons to be small sample size and choice of the splitting. The *scaffold* splitting, while being the most effective to establish an out-of-distribution generalisation performance, can also lead to the instability of the methods due to the structure of the molecules being very different among the training, validation and test set.

FUTURE WORK. Future research should focus on obtaining a deeper understanding of the negative transfer behaviour, investigating the task of interest in more details and comparing the distribution of source and target, with respect to input and learned features. The impact of varying the sample size in the source domain in also an interesting aspect to investigate. Overall, developing an insightful understanding of the graph properties, their topological structure and domain knowledge, will be decisive to ultimately establish well defined criteria for evaluating transferring capabilities. Another interesting future direction is to extend the current framework to other supervised learning problems on graphs, as node classification and link prediction. Finally, given the broad applicability and abundance of graphs, it will be exciting to explore the power of adversarial layers on different application domains, including signal and social networks or knowledge graphs.

Part IV

Summary and Outlook

7 Conclusions and outlook

The aim of this chapter is to summarize and discuss the main findings of our research, and then outline, in our critical opinion, the more promising directions to extend this work.

At the beginning of this thesis we described the complexity of the human brain, presenting some of the learning tools and ideas that researchers have investigated to gain a better understanding of it. While this ambitious goal is still far from being achieved, by now we have moved a few steps forward in this process.

7.1 Further exploration of neurological tasks

In Chapters 3 and 4 we performed an extensive analysis of MRI data with respect to various modalities and tasks. Our study offers a unique perspective in terms of sample size and images availability. Indeed, compared to most of the previous neuroimaging literature, our cohort has a larger number of subjects and benefits from a wide variety of imaging modalities. Nevertheless, from a machine learning perspective the sample size is still too small, leading to several issues as we will illustrate below.

Detect brain activity associated with the disease. In the MDD application, we analysed a population of 118 individuals. In neuroimaging, such a large cohort is usually obtained via meta-analysis, i.e. combining different studies, hence leading to obvious challenges in terms of data alignment, acquisition protocol and preprocessing [189]. Within our work, we were able to detect patterns of activity in different brain areas associated with the disease, while resolving some of the conflicting findings from the literature. At the same time, we obtained satisfying predictive performance when distinguishing between patients and controls. Possibly, the most relevant limitation of our study is the heterogeneous nature of depression as a disease itself. It is known that different MDD subtypes result in different effects on the cognitive process. Furthermore, some of the subjects in our population were under medication, possibly causing brain alterations both at the structural and functional level [66]. While these problems are partially addressed within the MVPC approach, which inherently takes into account confounding factors and noise in the data, future studies should investigate the effect of medication and disease subtype on brain activation patterns. Although this aspect was not considered for the MDD vs control task, our subsequent study addressed the related problem of treatment response prediction in the patient group. In this case, we were able to identify a particular brain region (aPHCr) that was mostly involved in the

response prediction task, while obtaining significant classification accuracies. However, since structural data were used for the analysis, the brain area of interest should be interpreted as informative of a structural alteration rather than an activation pattern. An interesting next step would be to combine the two studies to detect functional alterations as an effect of treatment or integrate the treatment effect as a controlling factor in the MDD phenotype prediction task.

IMPROVED MODALITY INTEGRATION. While we have not yet investigated task interactions so far, we examined the problem of imaging integration. Previous work has found a positive outcome from MRI data fusion in depression, as reported in a recent review [59]. In our cohort, the findings are not so clear. On one hand, we observe that specific subset of modalities combination can improve over their individual counterparts. On the other hand, we should note that the improvement was not consistent across data, with the performance often being dominated by the best single modality. Additionally, our analysis should be interpreted in retrospect, considering that only a selected subset of the possible modalities was tested, without accounting for the multiple comparison problem. We speculate a possible reason for our limited improvement to be the small sample size. Indeed, we used a whole-brain voxel space, resulting in the number of features being much larger than the number of samples. Previous work often used region based features, reducing the search space but also loosing information. Hereby, it is not surprising that, contrarily to our case, in such a scenario a combination of multiple data source could provide additional insights. In our setting the single features were informative enough to detect the brain signal, therefore adding information could be redundant, if not harmful. Similar considerations also apply to the multiple sclerosis study. Despite using region based features, the combined data metrics were all extracted from the same diffusion images, hence providing related information that a machine learning model might be incapable to pick in a small sample size regime. Then, the natural next step for investigation should be on the feature architecture side. Providing input information that has already been optimized for the task of interest, would be particularly advantageous for a multi-modal approach. We envision that supervised and unsupervised deep learning based approaches will play a major role in this context, given their undeniable ability to generate meaningful embeddings [80].

FEATURES FOR COMPLEX TASKS. Finally, we observed that complex tasks in MS are extremely challenging due to the unpredictable course of the disease. We believe that extracting relevant features is even more important in this context, considering that only with a bare eye observation of the image it is often impossible to identify the progressive status of the disease. Again, deep learning models have a great potential to improve the feature learning step, by capturing hidden interactions invisible to the human eye [173].

7.2 Extending Wasserstein kernels

Chapter 5 introduced the Wasserstein Weifeiler–Lehman kernel, overcoming two of the major limitations of previous work: (1) the simplicity of the aggregation step in the $\mathcal{R}$ – convolution framework; (2) the lack of generalization to graphs with continuous node attributes. While our method successfully addressed these shortcomings, we envision several next steps for extension and improvement of our work.

Edge attributes. Our setup did not explicitly account for high–dimensional edge attributes. However, with proper adjustments, the propagation scheme we defined in equation 5.10 can include edge attributes of arbitrary dimension. The easiest solution, would be to aggregate the high dimensional array into a single value, then treat it as an edge weight. A more interesting option would be to use the *dual* graph, constructed by reversing the node-edge representation. In the dual graph, nodes are the edges of the primal (original) graph; the latter are connected if the corresponding edges share a node in the primal graph. Ultimately, one could apply the WWL node propagation scheme on the primal and dual graph, then combine the kernels by appropriate weighting.

Positive definiteness in the continuous case. From a theoretical perspective, the main challenge of our approach is the lack of proof for the positive definiteness of the WWL kernel in the continuous setting. The considerations in Section 5.3.1 lead us to speculate that, under certain conditions, the WWL can be proved to be positive definite, as also supported by the experimental analysis. The complementary arguments rely on the observation that the space created via the Graph Wasserstein Distance is locally flat. While we were not able to present a complete proof yet, we hypothesize that this holds if the dimensionality of the feature space does not explode. Indeed, we conjecture the existence of a theoretical bound depending on the dimensionality and on the features scale. We certainly encourage future research to pursue this direction and formalise our high-level discussion in searching for a definitive proof. Although recent workaround to account for indefinite kernels have gained increasing interest in the community, most of the well established algorithms assume the Gram matrix to be positive definite. Therefore, guaranteeing this condition is a crucial step to extend the applicability of our method on new domains.

Speed-up and extension to neural networks. Another important discussion point is in the scalability of the WWL. In our data sets with small graphs we observed that the computation of the Wasserstein distance is still tractable, especially as this is a one-time operation for the algorithm. However, it would be appealing to be able to extend the approach on different data domains, such as social networks, or to different prediction tasks, for example node classifications or link prediction. These kind of applications normally deal with very large graphs, with hundreds of thousand or even million of nodes, for which the Wasserstein distance would be infeasible to compute in the classical implementation. We empirically tested the impact of using approximation

algorithms that speed-up the computation of the Wasserstein distance (see Section 5.4), indeed observing the more benefits as the number of nodes in the graph grows. While using these tricks can lead to a slight drop in predictive performance, these algorithms can be extremely useful to guarantee a broad applicability of our method. In the literature, the Wasserstein distance has been successfully applied as a loss function in neural networks and generative models [5, 56]. Combining this idea with our method could lead to novel Wasserstein based graph neural networks, whose development is still at an early stage [122]. Besides, developing a Wasserstein based GNN inspired by our propagation scheme would also facilitate applicability on large graph settings.

WASSERSTEIN KERNELS ON DIFFERENT DATA STRUCTURES. Lastly, we observed that Wasserstein distances can be exploited to create kernels on a variety of data structures. We already performed a pioneer work following this direction on the time series domain [17]. In our work, we defined a Wasserstein distance on subsequences of the time series to create effective similarity measures for classification. We showed that, as for the graph application, our approach overcomes the limitations of the $\mathcal{R}$ –convolution framework, which in the time series context degenerates into a simple comparison of their means. Our competitive experimental results on benchmark data sets [33] emphasize the importance of using optimal transport theory to simultaneously capture local and global characteristics of the data. We are confident that Wasserstein inspired kernels would be beneficial in many other application domains, such as strings or images, as preliminary research is already suggesting [39].

7.3 PERSPECTIVES IN DOMAIN ADAPTATION

Our ADV-GNN model and its variants represent a pioneer work in the field of adversarial learning on graphs, for the application on supervised graph–level classification. In this section we discuss some of the related work and outline both extensions and limitations of our method with respect to the existing literature.

TOWARDS MODEL IMPROVEMENT. Given the unique perspective of our approach, we envision the existence of almost limitless possibilities in terms of architectural improvements to increase the efficiency of our method. Related work focused on revising the feature learning representation step to encourage similar embeddings on the two domains. This is achieved either by minimising the distance between source and target distribution [113, 114], or employing a generative component [85]. A first incremental step would be to integrate these ideas within our framework, with the simple action of including an additional term in the loss function. Another interesting exploration would establish the impact of node and graph-level transfer, in a similar fashion as Hu *et al.* [89], where they showed that a combined methodology yields the more competitive results. A similar rationale could be applied on the adversarial layer component, by devising a multi-layer architecture jointly discriminating embeddings at the node and graph level. Moreover, including a node-level adversarial layer would facilitate

applications on different tasks, such as link prediction or node classification. Despite domain adaptation for node classification was already investigated, our view would substantially differ from previous work [195]. Wu *et al.* [195] studied the case of transferring to unlabelled target data from large labelled source domain, while we are considering the scenario of limited labelled target data with unlabelled source. Besides, given the versatility of graph based representations our approach could be extended on different application areas, for example text classification, as already explored [194]. The potential success of our method on different domains is already confirmed by previous work, where a task-shared based architecture was devised for image classification problems [24]. In fact, the idea of explicitly separate the learned latent features into domain versus class specific is well known in image classification, where it has shown to clearly improve over the naive transfer learning techniques [140].

GRAPH DISTRIBUTION. One of the most critical discussion points in any transfer learning setting is the type of relation between distribution of source and target. Ideally, the most similar they are the better the features can be shared across domains. In our empirical evaluation, we chose all data sets containing small molecules, sometimes also screened for related properties. Hereby, we consider this similarity to be sufficient for the transfer learning task. However, formally evaluating the alignment between distributions, and eventually filter outlier samples, should be a common research practice. Nonetheless, for graphs it is not so trivial to establish a meaningful measure of domain similarity. Indeed, we believe that our graph Wasserstein distance could further be employed for this task.

7.4 A UNIFIED FRAMEWORK FOR GRAPHS IN BRAIN MRI

In MRI data analysis, the feature extraction step is possibly the most relevant to the subsequent prediction. Depending on the task of interest, one can derive a multitude of different information from the scan. In our work we focused on image related features, either high-resolution (voxel-by-voxel) or low-dimensional region of interest. Nevertheless, to capture the complexity of the brain one can exploit more involved extraction pipelines, leading to a representation with different data structures. In the introduction of this work, we argued how graphs are ideal candidates to represent the complexity of the brain, given their flexibility in the type of information they can store. We observed that, at an high-level, one can distinguish between functional and structural networks (see Section 1.3 and Figure 1.3). In particular, each of the MRI modalities that we considered, structural, functional and diffusion MRI, lead to different type of graphs [27, 79].

CONSTRUCT THE BRAIN GRAPH. In general, nodes are defined either at a voxel level (i.e. one node per voxel) or at a region level (i.e. one node per brain area). In the latter case, the regions could be determined by well established anatomical masks. User customized masks can also be created, for example using a fixed radius or number of

voxels within the region, ultimately resulting in a total brain parcellation. With respect to the graph edges, the way they are defined is strictly dependent on the MRI modality.

1. Structural MRI: anatomical connectivity is inferred by looking at the covariance of morphological measures, such as volume or cortical thickness.
2. Diffusion MRI: the connectivity is defined via the probability of existing white matter tracts between pair of grey matter regions.
3. Functional MRI: measure of statistical correlation between regions of the brain, based on the time series BOLD signal response.

For fMRI images, most commonly, networks are obtained from resting state data. In the task-based experiments, it is not so trivial to define correlation measures, given that the time series consists of blocks of related stimuli. Then, one should consider the different trials independently and subsequently aggregate the results or employ new specific measures to look at task-related synchronization [112]. Possibly, the trickiest component to create the brain network is in the threshold definition for edge selection. In principle, both at the structural and functional level, using the procedure described one could obtain (almost) fully connected graphs. For example, the correlation based fMRI matrix can be obtained for every pair of regions or voxels, resulting in the fully connected graph. Nevertheless, for an interpretable representation, we aim to reduce the edge density, so that only the significant connections are displayed. This is achieved by introducing a threshold and remove edges which do not meet the required criteria. How to define the threshold is still an active research question: typical approaches use customized, statistical or expert based criteria. For instance, one can choose to only keep the edges that survive a statistical significant assessment at the group level (SPM; see Section 3.2.1). Another option would be to employ a cross validation based approach, testing different threshold values on independent set of images and select the most promising with respect to the task of interest. User defined techniques mostly rely on findings from previous literature, or comparing the network to well-known default connections in an healthy population [27].

GRAPH ANALYSIS. With respect to the subsequent graph analysis of brain MRI data, we identify two main perspectives: (1) topological and statistical; (2) graph modelling. Intuitively, the topology is a characterization of the shape of the graph. This is determined, for example, by nodes degree, edge density, shortest path length or, more generally, by connectivity and efficiency measures. Once the topological properties have been defined, one can compare individual brain graphs among subjects from different groups, or with respect to established brain networks. The topological measures can also be employed as input features for a classifier. From a statistical point of view, inference between extracted topological and general graph properties can be performed, to assess the statistical difference across networks.

The most interesting direction in the perspective of this thesis is the graph modelling application, which would include graph kernels and graph neural networks as the major methodological tools. Remarkably, earlier work already employed the WL kernel

for classification of tasked-based fMRI [180, 181]. Nevertheless, these studies did not include a clinical phenotype prediction problem, but rather aimed to predict the type of stimuli from the voxel time series. Later on, this idea was extended by Zhou *et al.* [204] who tried to distinguish between Schizophrenic subjects from healthy controls using WL graph kernels on resting state networks. We performed a similar preliminary analysis on our cohort which did not result in an outstanding outcome. However, this unsatisfactory performance could be due to the limited sample size or to the image processing step. Therefore, we strongly encourage further exploration in this direction possibly leading to a major breakthrough in graph–based MRI analysis. Furthermore, previous work could not account for edge weights, given the inability of the classical WL to include them. As a consequence, we speculate that WWL will be an improved and more versatile solution to solve graph classification problems in the MRI domain. Graph neural networks would also be an extremely interesting methodology and research direction to pursue for the particular MRI application. To date, this area of investigation has done very little progress, presumably due to the lack of imaging data which is a major obstacle for deep learning approaches. Nonetheless, some pioneer studies have explored the development of GNNs for various clinical prediction task of MRI networks, reporting promising initial results [110, 121, 191].

ROBUSTNESS OF THE GRAPH. One of the main limitations when dealing with brain graphs is the uncertainty of the underlying structure and features. As we mentioned, there are no clearly defined approaches to choose appropriate thresholds, which ultimately define the graph adjacency matrix itself. Besides, also node and edge features are arbitrarily defined, being either categorical location arguments, tissue characteristics, or functional activation. Therefore, employing methodologies that are robust with respect to graph perturbation would certainly guarantee more reliable results. Graph kernels represent ideal candidates from this perspective, since their robustness to node and edge alterations has been studied. The empirical findings showed that, under small noise perturbations, the classification performance does not drastically drop [132, 199]. In parallel, robustness of graph neural network is mostly an unexplored topic. Nevertheless, recent efforts have been made to equip existing GNNs with robust training and regularization techniques, with the aim to improve robustness [19, 170, 205]. Further studies focusing on the identification of criteria for assessing perturbation robustness in graph neural network models, would certainly provide additional inspiration for applied researchers to employ these methods on MRI data.

7.4.1 UNDERSTANDING DOMAIN ADAPTATION FOR MRI

In Chapter 6 we presented adversarial graph neural networks, a family of approaches to extend transfer learning techniques on the graph domain via adversary layers. Transfer learning is mostly required for model improvement on small sample size and limited data domains, then it is also an efficient strategy to overcome data related shortcomings in MRI. Indeed, the successful results obtained in classical imaging analysis suggest that a similar pattern of improvement could be observed on brain MRI scans. Within our

multiple sclerosis and depression study, we already emphasized the uncertainty of the phenotype. A wrongly assigned class label not only results in suboptimal models, but further contributes to create a label distribution which is different from the real one. In this scenario, one could turn the problem into a semi–supervised task and employ domain adaptation techniques to learn on a large source and predict on the small target domain with missing label information. Another challenge faced in the MDD study came from the images being acquired at multiple acquisition sites. As it is problematic to exactly reproduce the same conditions in different sites, MRI studies often present intrinsic data discrepancies within the same cohort. Again, transfer learning could be extremely valuable in these cases, where the site condition can be explicitly taken into account to derive shared features representation across domains. Of course, the problem of getting reliable annotated source data sets still remains. Luckily, a lot of work was done in recent years to create databases and biobanks supporting researchers in the data collection [127, 176]. With these resources keep growing, we expect that transfer learning will shape the future of MRI data analysis in the coming years.

TRANSFER LEARNING ON BRAIN GRAPHS We conclude this section outlining the perspective integration of all the ingredients we developed so far in a unifying framework. On one hand, obtaining improved and robust graph based representation of brain MRI data opens the path to multiple opportunities. Mostly, one could gain benefit from the large progress of graph based machine learning models. Graph kernels and graph neural networks have been widely developed in the last two decades and this grow will likely continue in the future. On the other hand, limitations in MRI data and the small sample size regime make many of these techniques suboptimal. Therefore, transfer learning plays the crucial role of linking the input brain MRI data with machine learning classification methods, and particularly graph based approaches. Sharing knowledge across MRI studies and domains will improve the learning algorithm capabilities and optimize the feature space to the limited target data set, hereby resulting in higher performance.

7.5 CONCLUSION

It is undeniable the enormous effect that machine learning and artificial intelligence are having across a variety of application fields, with no signs that this race will stop any time in the near future. Despite this exponential technological growth, there are still many open challenges for researcher and practitioners to address. The clinical domain is certainly among the most fascinating, due to the clear relevance and direct impact on our life. In this work, we discussed how the low sample size limitation is a non negligible burden for machine learning models, strongly affecting their learning capabilities. Model interpretation is another crucial aspect in the medical field. To this end, effective collaborations between data analysts and clinicians are almost mandatory in any healthcare related study, above all on MRI, where expert knowledge from radiolo-

gists and engineers is also required to ensure correctness of acquisition and processing protocols.

In this thesis we tackled some of these challenges from a broad perspective, discussing the relevance of structured graph data and MRI analysis, to ultimately enable knowledge transfer on limited biomedical domains. We believe that research in these area will continue to expand. The development of methodologies to translate across domains will be crucial in this context, as part of a fully integrated system comprising software, analysis, and medical databases to handle this knowledge explosion.

A MAGNETIC RESONANCE IMAGING: MODALITIES, ACQUISITION, PREPROCESSING, AND ANALYSIS

A.1 MAGNETIC RESONANCE IMAGING

Magnetic resonance imaging (MRI) is a non-invasive medical imaging technique that uses a magnetic fiend to create 3 D images of the body [1]. We are particularly interested in brain MRI, that is MRI images of the human head. Depending on the signal acquisition technique and scanner setup different modalities of brain MRI data can acquired, providing a diverse type of information of the anatomical and functional activity of the brain.

A.1.1 STRUCTURAL MRI

Structural MRI (sMRI) is especially used to represent the anatomy of the brain, including shape and size, but can also provide a tissue separation into White Matter (WM), Gray Matter (GM) and Cerebrospinal Fluid (CSF). The magnetic field producing the MRI signal is characterized by the pulse frequency, which determined the time between the input is delivered and he signal reception. Varying the pulse and the length of acquisition, determine a different contrast of the image and therefore an emphasize different characteristics. The most common sequences in MRI are T1 -weighted and T2-weighted: the first one provides a good contrast between GM and WM, while the latter one between brain tissue and CSF [2].

DIFFUSION MRI. Diffusion weighted imaging (DWI) [52] is also a structural technique, however contrarily to the T1 -weighted sMRI instead of depicting the standard anatomy of the brain, it aims at detecting movements in water molecules within the brain. More precisely, DWI collects a sequence of T2 -weighted images, and applying gradient pulses in the 3 orthogonal directions derives the water diffusion paths from a measure of tissue density. An hypointense diffusion signal is a sign of tissue damage, and in particular of the white matter tracts in the brain; consequently, diffusion MRI is particularly suitable to detect disease that structural brain damage such as ischemia, acute stroke. or demyelination, which is also associated with multiple sclerosis.

[1] Source: https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768

[2] Source: https://cfmriweb.ucsd.edu/Howto/3T/structure.html

A.1.2 Functional MRI

Functional MRI is a neuroimaging technique which measures the activity in the brain by detecting changes associated with blood flow [3] [151]. This is achieved by looking at the blood-oxygen-level dependent (BOLD) contrast, that can be measured within the scan by exploring variations in the hemodynamic response; in practice, the response is evaluated by looking at the blood flow signal and its rapidity, which is associated with the ability of a subject to respond to specific stimuli [90].

Resting-state fMRI. Resting-state fMRI (rs-fMRI) is type of functional image, where the BOLD signal is recorded when the subject is a at rest, implying that no task is performed. This is used to identify inherent and physiological brain activity patterns, and often used to map the functional brain network structure [134].

Task-based fMRI. In the task-based fMRI the subject performs a give task during the scanner session. These tasks are typically related to motor, auditory and visual stimuli, for example looking at a group of images or listening to music. The stimuli are shown in sequence and often a block designed is used, such that contrast measures between sequence of images within a block can be derived for subsequent analysis.

A.2 Major Depressive Disorder study

A.2.1 Data acquisition and preprocessing

Task-based fMRI

At FUB, functional data were acquired with 37 oblique axial slices of 3 mm (field of view 192 mm, 3×3 mm in-plane, repetition time 2s or 2.3 s, echo time 30 ms, flip angle 70°). At UZH functional data were acquired using a sensitivity-encoded single-shot echo-planar sequence (TE = 35 ms; field of view = 22 cm; acquisition matrix = 80×80, interpolated to 128×128, voxel size = $2.75 \times 2.75 \times 4$ mm, and sensitivity-encoded acceleration factor R = 2.0) sensitive to blood oxygenation level-dependent (BOLD) contrast (T 2^* weighting). Using a midsagittal scout image, 32 contiguous axial slices were placed along the anterior–posterior commissure plane covering the entire brain with a repetition time of 2000 ms ($\theta = 82°$). During preprocessing, functional data were registered to the mean, corrected for motion artefacts, mean-adjusted by proportional scaling, normalized into standard stereotactic space (template provided by the Montreal Neurological Institute), and spatially smoothed using a 6 mm FWHM Gaussian kernel. The time series were high-pass filtered to eliminate low-frequency components (filter width 128 s) and adjusted for systematic differences across trials. Single subject analysis on the preprocessed fMRI data was performed by modelling the different conditions (Fixation, Negative, Neutral, Positive) convolved with a hemodynamic response function as explanatory variables within the context of the general linear model

[3] https://en.wikipedia.org/wiki/Functional_magnetic_resonance_imaging

on a voxel-by-voxel basis. Realignment parameters were included as additional regressors in the statistical model.

RESTING-STATE FMRI

The resting state data at FUB were collected in 8 minute runs (210 vol) . At UZH the functional images were collected in 10 min runs (200 vol) with 32 contiguous axial slices of 4 mm and a repetition time of 3000 ms. All the other parameters are analogous to the task-based fMRI data. The resulting residual BOLD time series were further band-pass filtered ($0.01 - 0.1$ Hz).

STRUCTURAL MRI

The images were acquired using a standard quadrature head coil (TR = 1900 ms; TE = 2.52 ms; flip angle = 9°; 176 contiguous sagittal slices; field of view = 256 mm; acquisition time 4 : 26 min). The raw and preprocessed images were individually inspected for artifacts and image quality; in addition, all scans passed through an automated quality check protocol. None of the analysed images showed abnormalities. The normalized GM maps were smoothed with an isotropic Gaussian kernel (FWHM = 8 mm).

A.2.2 ADDITIONAL RESULTS

BETA IMAGES. We report the complete results with the different *Beta* images of a C-SVM with various kernels: polynomial with degree 2 and 3 and sigmoid (Figures A.1, A.2, A.3). We observe a very similar pattern as for the linear and RBF examples reported in the main text: unsurprisingly, the break stimuli image (B 04) tends to give lower results and no clear benefit is observed by combining an average signal of the 4 images.

CONTRAST IMAGES. Being aware of the limitations of a leave-one-out approach in terms of robustness and stability [178] we additionally repeat the experiments of SVM-fScore on the contrast images using a 10 fold CV. The results reported in Table A.1 corroborate the validity of our findings, despite the predictive performances are slightly lower than the LOO approach; this is not surprising given the limited sample size availability.

A.3 MULTIPLE SCLEROSIS STUDY

A.3.1 DATA ACQUISITION

The MRI protocol consisted of single-shell diffusion weighted MRI (b = 1200 s/mm^2 61 gradient directions; resolution $2 \times 2 \times 2$ mm; TE = 68 ms; TR = 24000 ms, depending on the cardiac gate), while MT data included MT ”on” and MT "off" images. Standard anatomical 3 D T1 -weighted gradient echo and 2 D PD-T2 -weighted turbo spin echo

Figure A.1: Performance of multiple Beta images with polynomial kernels of degree 2.The left plot show accuracy precision and recall for each Beta and for the average image. On the right side, the ROC curve is reported.

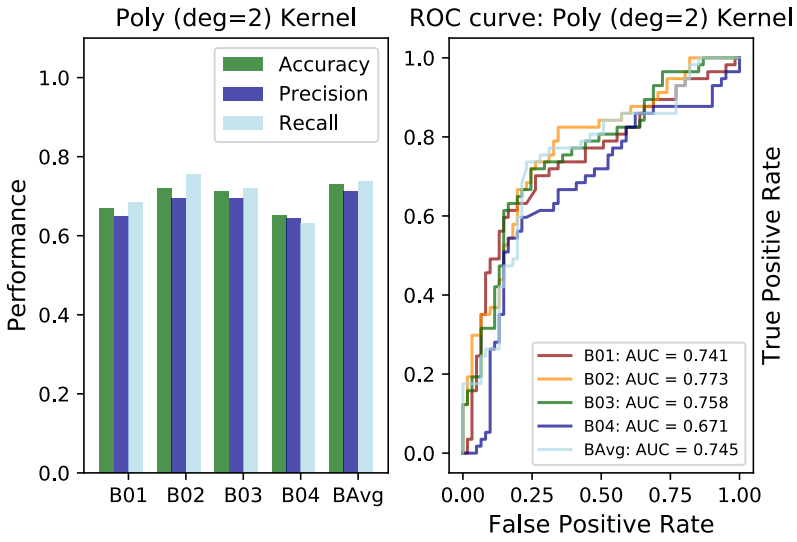


Table A.1: Classification results on the contrast of interest with SVM-fScore and 10 fold CV.

Contrast	Accuracy	Sensitivity	Specificity
WM > Break	64.24 ± 2.36	68.42 ± 1.92	60.33 ± 3.80
Pos > Break	62.54 ± 1.96	65.61 ± 3.25	59.67 ± 2.22
Neg > Break	62.88 ± 1.64	65.26 ± 4.49	60.66 ± 4.27
Neu > Break	68.14 ± 2.60	71.93 ± 2.48	64.59 ± 3.04
Emo > Neu	50.51 ± 2.66	75.44 ± 2.94	27.21 ± 3.38

images were also acquired as part of the protocol in all subjects and time points, as this would enable detailed tissue segmentation and MS lesions delineation.

Figure A.2: Performance of multiple Beta images with polynomial kernels of degree 3.The left plot show accuracy precision and recall for each Beta and for the average image. On the right side, the ROC curve is reported.

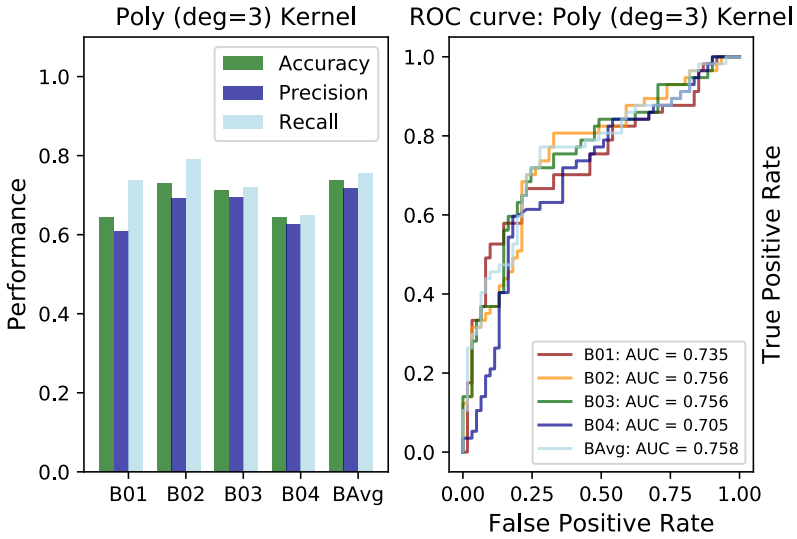
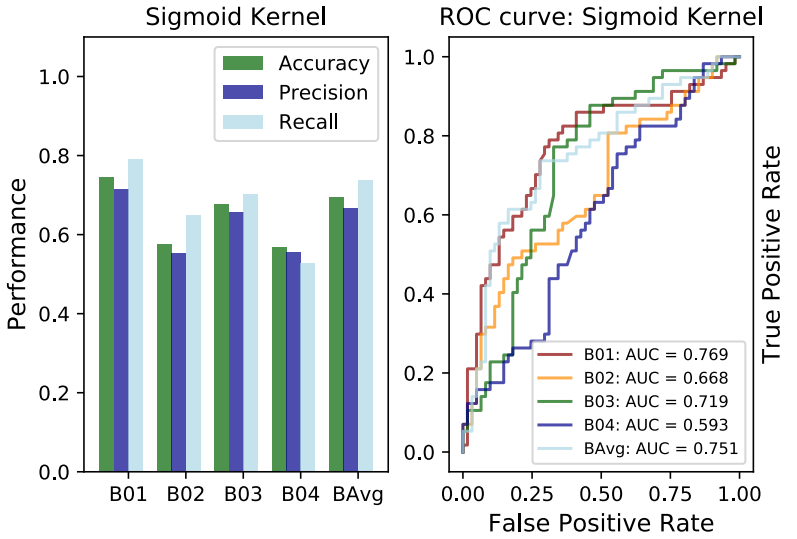


Figure A.3: Performance of multiple Beta images with a sigmoid kernel.The left plot show accuracy precision and recall for each Beta and for the average image. On the right side, the ROC curve is reported.



B SOFTWARE AVAILABILITY

B.1 A CLINICAL DECISION SYSTEM FOR MRI DATA: SOFTWARE INTEGRATION

As part of the Clinical Decision System of the CDS-QuaMRI[1] Horizon 2020 project, we aim to integrate our classification modules implemented for the MRI analysis into a unified software framework. The system has been developed at GyroTools[2] and already includes several modules for processing of quantitative MRI data. With respect to the classification module, we rely on three main tasks: (1) data upload; (2) feature extraction; (3) model training. The classification module is a central application in the clinical decision support system, which relies on all other modules to perform feature extraction from the different labelled modality data, training of a classifier using machine learning principles, and finally applying the trained model for prediction on unlabelled data.

B.1.1 DATA UPLOAD

As it became clear during the implementation of the project, the system should allow to handle multiple classification strategies, as no unified technique will be applicable to a broad variety of input data. Also, due to increased regulatory constraints regarding data safety and privacy, centralized data collection from centres in different institutions and countries has become virtually impossible. As a consequence, the requirements for the framework implementation had to be expanded to allow for decentralized processing of data and decentralized training of the classification model. This is also a setup aligned with the user, who will be able to upload the MRI data and create a personalised data base.

B.1.2 FEATURE EXTRACTION

The feature extraction is fully integrated as a task in the system framework and developd in `Python`. The adopted solution takes a set of image data in `Nifti` format as input. Then, data of different formats is converted and the output is a vector of features (.npy format), to feed to the database system for optional storing. Features are then aggregated over samples to create the input data matrix for the model training module. The feature extraction pipeline can be summarized as:

[1] https://cds-quamri.eu/

[2] https://www.gyrotools.com/gt/

- Create mask. Input: single subject (directory containing Nifti image). Output: mask (Nifti file)
- Create Features. Input: single subject (directory containing Nifti file) and a mask (Nifti file). Output: array of features (.npy)
- Create Data Matrix. Input: list of subjects (directories containing Nifti file) and mask. Output: data matrix (.npy)

B.1.3 Model training

The training of the model is also realized in `Python` using publicly available machine learning libraries. The input is the data matrix generated in the previous step. Output are the model parameters as refined by the training algorithm. The model is then stored in the database and can be applied on other data, for example to determine classification performance on a test sample. At the moment, a prototype of a Support Vector Machine model has been used, given the successful performance obtained in Chapters 3 and 4. The model training pipeline can be summarized as:

- Train model. Input: data matrix. Output: model parameters
- Save model. Input: trained model parameters. Output: binary model file

B.1.4 System integration

The classification module is entirely realized as a task in the overall framework. The framework allows for exchange of tasks between sites with different installations of the framework. Also the models can be exchanged between installations, which allows to refine a partially trained model with additional data at a different site or institution.

B.2 A software framework to compute graph kernels

One of the key issued in the graph kernels field is reproducibility. Section 2.2 introduce a variety of different approaches, which is not even inclusive of all the available methods [103]. Overall, there is no agreement about the benchmark data set used, training and validation splits, or hyperparameters selection, leading to inconsistent empirical results across publications. Furthermore, the lack of published code, or implementation in different programming languages, is a major obstacle in establishing a common experimental setting within the community. To address this problem, public software packages that facilitate the application and implementation of graph kernels in popular and uniform coding languages have recently been developed [162, 168]. Our contribution, is the `graphkernels` package, including `Python` and `R` libraries relying on an efficient `C++` backend implementation [168]. The user-friendly interface permits the computation of individual kernel matrices with only a few lines of code. Furthermore, the similar interface between the `Python` and `R` versions, facilitate the user with versatility across the two languages. `graphkernels` supports 14 kernels from the following families:

(i) Graph kernels between node and/or edge histograms:
- Linear: `VertexHist`, `EdgeHist`, `VertexEdgeHist`, `VertexVertexEdgeHist`
- Gaussian RBF: `VertexHistGauss`, `EdgeHistGauss`, `VertexEdgeHistGauss`

(ii) Graphlet kernels: `Graphlet`, `ConnectedGraphlet`

(iii) Random walk kernels: `KStepRandomWalk`, `GeometricRandomWalk`, `ExponentialRandomWalk`, `ShortestPath`

(iv) The Weisfeiler-Lehman subtree kernel: `WL`

B.2.1 How to use `graphkernels`

The user interface is very simple: given a collection of graphs $G_1 \ldots, G_n$, the kernel matrix $K \in \mathbb{R}^{n \times n}$ is returned with respect to each kernel. Here, we illustrate an example usage of the `Python` package using the benchmark dataset MUTAG [40], which is also provided with the installation.

1. Load the required packages. Import the *graphkernels* library and *numpy* library.

```
>>> import graphkernels.kernels as gk
>>> import numpy as np
```

2. Load the data.

```
>>> # Load the data in the graphkernels package folder
>>> data = np.load("graphkernels/data.mutag")
```

3. Compute the kernel matrix with WL

```
>>> K = gk.CalculateWLKernel(data, 5)
```

computes the WL kernel k_{WL} for the parameter $h = 5$, corresponding to the number of WL iterations.

Additional examples and details on the implemented kernels can be found in Sugiyama *et al.* [168] and in our `GitHub` repository[3].

[3] https://github.com/BorgwardtLab/GraphKernels

Bibliography

1. F. Aiolli and M. Donini. "EasyMKL: a scalable multiple kernel learning algorithm". *Neurocomputing* 169, 2015, pp. 215–224.
2. F. Aiolli, G. Da San Martino, and A. Sperduti. "A kernel method for the optimization of the margin distribution". In: *International Conference on Artificial Neural Networks*. Springer. 2008, pp. 305–314.
3. Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson. "Deep learning for brain MRI segmentation: state of the art and future directions". *Journal of digital imaging* 30:4, 2017, pp. 449–459.
4. J. Altschuler, J. Niles-Weed, and P. Rigollet. "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *Advances in neural information processing systems*. 2017, pp. 1964–1974.
5. M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein gan". *arXiv preprint arXiv:1701.07875*, 2017.
6. J. Ashburner. "A fast diffeomorphic image registration algorithm". *Neuroimage* 38:1, 2007, pp. 95–113.
7. J. Ashburner and K. J. Friston. "Unified segmentation". *Neuroimage* 26:3, 2005, pp. 839–851.
8. W. M. Association. "World Medical Association Declaration of Helsinki: Ethical Principles for Medical Research Involving Human Subjects". *JAMA* 310:20, 2013, pp. 2191–2194.
9. T. Y. Azizov and I. S. Iokhvidov. "Linear operators in spaces with an indefinite metric and their applications". *Itogi Nauki i Tekhniki. Seriya" Matematicheskii Analiz"* 17, 1979, pp. 113–205.
10. L. Babai and L. Kucera. "Canonical labelling of graphs in linear average time". In: *20th Annual Symposium on Foundations of Computer Science*. 1979, pp. 39–46.
11. P. Baldinger, A. Lotan, R. Frey, S. Kasper, B. Lerer, and R. Lanzenberger. "Neurotransmitters and electroconvulsive therapy". *The journal of ECT* 30:2, 2014, pp. 116–121.
12. A. T. Beck, R. A. Steer, G. K. Brown, *et al.* "Beck depression inventory-II". *San Antonio* 78:2, 1996, pp. 490–498.
13. A. T. Beck, R. A. Steer, R. Ball, and W. F. Ranieri. "Comparison of Beck Depression Inventories-IA and-II in psychiatric outpatients". *Journal of personality assessment* 67:3, 1996, pp. 588–597.

14. R. Bellman. “Dynamic programming”. *Science* 153:3731, 1966, pp. 34–37.
15. C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic analysis on semigroups: theory of positive definite and related functions*. Vol. 100. Springer, 1984.
16. C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
17. C. Bock, M. Togninalli, E. Ghisu, T. Gumbsch, B. Rieck, and K. Borgwardt. “A Wasserstein Subsequence Kernel for Time Series”. In: *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2019, pp. 964–969.
18. J. Bognár. *Indefinite inner product spaces*. Vol. 78. Springer Science & Business Media, 2012.
19. A. Bojchevski and S. Günnemann. “Certifiable Robustness to Graph Perturbations”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 8319–8330.
20. K. Borgwardt. *Graph Kernels*. Ludwig-Maximilians-University Munich, 2007.
21. K. M. Borgwardt and H.-P. Kriegel. “Shortest-path kernels on graphs”. In: *Fifth IEEE international conference on data mining (ICDM'05)*. IEEE. 2005, 8–pp.
22. K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel. “Protein function prediction via graph kernels”. *Bioinformatics* 21, 2005, pp. i47–i56.
23. B. E. Boser, I. M. Guyon, and V. N. Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. 1992, pp. 144–152.
24. K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. “Domain separation networks”. In: *Advances in neural information processing systems*. 2016, pp. 343–351.
25. M. R. Bridson and A. Haefliger. *Metric spaces of non-positive curvature*. Vol. 319. Springer Science & Business Media, 2013.
26. J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. “Spectral networks and locally connected networks on graphs”. English (US). In: *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*. 2014.
27. E. T. Bullmore and D. S. Bassett. “Brain graphs: graphical models of the human brain connectome”. *Annual review of clinical psychology* 7, 2011, pp. 113–140.
28. M. J. Cardoso, M. Modat, R. Wolz, A. Melbourne, D. Cash, D. Rueckert, and S. Ourselin. “Geodesic information flows: spatially-variant graphs and their application to segmentation and fusion”. *IEEE transactions on medical imaging* 34:9, 2015, pp. 1976–1988.
29. S. Carney and J. Geddes. *Electroconvulsive therapy*. 2003.
30. R. Caruana, S. Lawrence, and C. L. Giles. “Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping”. In: *Advances in neural information processing systems*. 2001, pp. 402–408.

31. T. D. Challenge. *Tox21 data challenge 2014*. 2014. URL: https://tripod.nih.gov/tox21/challenge.

32. D. Chen, S. Liu, P. Kingsbury, S. Sohn, C. B. Storlie, E. B. Habermann, J. M. Naessens, D. W. Larson, and H. Liu. "Deep learning and alternative learning strategies for retrospective real-world clinical data". *NPJ digital medicine* 2:1, 2019, pp. 1–5.

33. Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. "The UCR time series classification archive", 2015.

34. Y.-W. Chen and C.-J. Lin. "Combining SVMs with various feature selection strategies". In: *Feature extraction*. Springer, 2006, pp. 315–324.

35. C. Cortes and V. Vapnik. "Support-vector networks". *Machine learning* 20:3, 1995, pp. 273–297.

36. M. Cuturi. "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems*. 2013, pp. 2292–2300.

37. C. Davatzikos. "Machine learning in neuroimaging: progress and challenges". *NeuroImage* 197, 2019, p. 652.

38. M. Davies, M. Nowotka, G. Papadatos, N. Dedman, A. Gaulton, F. Atkinson, L. Bellis, and J. P. Overington. "ChEMBL web services: streamlining access to drug discovery data and utilities". *Nucleic Acids Research* 43:W1, 2015, W612–W620. ISSN: 0305-1048.

39. H. De Plaen, M. Fanuel, and J. A. Suykens. "Wasserstein Exponential Kernels". *arXiv preprint arXiv:2002.01878*, 2020.

40. A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch. "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity". *Journal of medicinal chemistry* 34:2, 1991, pp. 786–797.

41. M. Defferrard, X. Bresson, and P. Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *Advances in neural information processing systems*. 2016, pp. 3844–3852.

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

43. E. W. Dijkstra *et al.* "A note on two problems in connexion with graphs". *Numerische mathematik* 1:1, 1959, pp. 269–271.

44. A. T. Drysdale, L. Grosenick, J. Downar, K. Dunlop, F. Mansouri, Y. Meng, R. N. Fetcho, B. Zebley, D. J. Oathes, A. Etkin, *et al.* "Resting-state connectivity biomarkers define neurophysiological subtypes of depression". *Nature medicine* 23:1, 2017, pp. 28–38.

45. D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. "Convolutional networks on graphs for learning molecular fingerprints". In: *Advances in neural information processing systems*. 2015, pp. 2224–2232.
46. A. Eklund, T. E. Nichols, and H. Knutsson. "Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates". *Proceedings of the national academy of sciences* 113:28, 2016, pp. 7900–7905.
47. J. S. Elam and D. Van Essen. "Human Connectome Project". In: *Encyclopedia of Computational Neuroscience*. Ed. by D. Jaeger and R. Jung. Springer New York, New York, NY, 2013, pp. 1–4.
48. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.* "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *KDD*. Vol. 96. 34. 1996, pp. 226–231.
49. A. Feragen, F. Lauze, and S. Hauberg. "Geodesic exponential kernels: When curvature and linearity conflict". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3032–3042.
50. A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. Borgwardt. "Scalable kernels for graphs with continuous attributes". In: *Advances in neural information processing systems*. 2013, pp. 216–224.
51. A. Figalli and C. Villani. "Optimal transport and curvature". In: *Nonlinear PDE's and applications*. Springer, 2011, pp. 171–217.
52. K. R. T. Fink, M. R. Levitt, and J. R. Fink. "Chapter 3 - Principles of Modern Neuroimaging". In: *Principles of Neurological Surgery (Third Edition)*. W.B. Saunders, 2012, pp. 53 –75.
53. R. W. Floyd. "Algorithm 97: shortest path". *Communications of the ACM* 5:6, 1962, p. 345.
54. J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.
55. K. J. Friston, A. Holmes, K. Worsley, J. Poline, C. Frith, and R. Frackowiak. "Statistical Parametric Maps in Functional Imaging: a General Linear Approach". *Human Brain Mapping* 2, 1995, pp. 218–229.
56. C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio. "Learning with a Wasserstein loss". In: *Advances in neural information processing systems*. 2015, pp. 2053–2061.
57. H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell. "Optimal assignment kernels for attributed molecular graphs". In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 225–232.
58. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. "Domain-adversarial training of neural networks". *The Journal of Machine Learning Research* 17:1, 2016, pp. 2096–2030.

59. S. Gao, V. D. Calhoun, and J. Sui. "Machine learning in major depression: From classification to treatment outcome prediction". *CNS neuroscience & therapeutics* 24:11, 2018, pp. 1037–1052.

60. B. Gaonkar, R. T. Shinohara, C. Davatzikos, A. D. N. Initiative, *et al.* "Interpreting support vector machine models for multivariate group wise analysis in neuroimaging". *Medical image analysis* 24:1, 2015, pp. 190–204.

61. A. Gardner, C. A. Duncan, J. Kanno, and R. R. Selmic. "On the definiteness of earth mover's distance and its relation to set intersection". *IEEE transactions on cybernetics* 48:11, 2017, pp. 3184–3196.

62. M. R. Garey and D. S. Johnson. *Computers and intractability*. Vol. 174. freeman San Francisco, 1979.

63. M. Gärtner, M. E. Ghisu, M. Scheidegger, L. Bönke, Y. Fan, A. Stippl, A.-L. Herrera-Melendez, S. Metz, E. Winnebeck, M. Fissler, A. Henning, M. Bajbouj, K. Borgwardt, T. Barnhofer, and S. Grimm. "Aberrant working memory processing in major depression: evidence from multivoxel pattern classification". *Neuropsychopharmacology* 43:9, 2018, pp. 1972–1979.

64. M. Gärtner, S. Aust, M. Bajbouj, Y. Fan, K. Wingenfeld, C. Otte, I. Heuser-Collier, H. Böker, J. Hättenschwiler, E. Seifritz, S. Grimm, and M. Scheidegger. "Functional connectivity between prefrontal cortex and subgenual cingulate predicts antidepressant effects of ketamine". *European Neuropsychopharmacology* 29:4, 2019, pp. 501–508.

65. M. Gärtner, E. Ghisu, A. L. Herrera-Melendez, M. Koslowski, S. Aust, P. Asbach, C. Otte, F. Regen, I. Heuser, K. Borgwardt, S. Grimm, and M. Bajbouj. "Using routine MRI data of depressed patients to predict individual responses to electroconvulsive therapy". *Experimental Neurology*, 2020, p. 113505.

66. T. Gärtner, P. Flach, and S. Wrobel. "On graph kernels: Hardness results and efficient alternatives". In: *Learning theory and kernel machines*. Springer, 2003, pp. 129–143.

67. C. Gaser and R. Dahnke. "CAT-a computational anatomy toolbox for the analysis of structural MRI data". *HBM* 2016, 2016, pp. 336–348.

68. A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit, and A. R. Leach. "The ChEMBL database in 2017". *Nucleic Acids Research* 45:D1, 2016, pp. D945–D954.

69. J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. "Neural message passing for Quantum chemistry". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 1263–1272.

70. R. Glem, A. Bender, C. Hasselgren, L. Carlsson, S. Boyer, and J. Smith. "Circular fingerprints: Flexible molecular descriptors with applications from physical chemistry to ADME". *IDrugs : the investigational drugs journal* 9, 2006, pp. 199–204.

71. M. M. Goldenberg. "Multiple sclerosis review". *Pharmacy and Therapeutics* 37:3, 2012, p. 175.

72. M. Gönen and E. Alpaydin. "Multiple kernel learning algorithms". *Journal of machine learning research* 12:64, 2011, pp. 2211–2268.

73. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

74. M. Gori, G. Monfardini, and F. Scarselli. "A new model for learning in graph domains". In: *2005 IEEE International Joint Conference on Neural Networks.* Vol. 2. IEEE. 2005, pp. 729–734.

75. S. Grimm, A. Weigand, P. Kazzer, A. M. Jacobs, and M. Bajbouj. "Neural mechanisms underlying the integration of emotion and working memory". *NeuroImage* 61:4, 2012, pp. 1188–1194.

76. M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase. "Generalized autocalibrating partially parallel acquisitions (GRAPPA)". *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 47:6, 2002, pp. 1202–1210.

77. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. "Gene selection for cancer classification using support vector machines". *Machine learning* 46:1-3, 2002, pp. 389–422.

78. B. Haasdonk and C. Bahlmann. "Learning with distance substitution kernels". In: *Joint pattern recognition symposium*. Springer. 2004, pp. 220–227.

79. M. G. Hart, R. J. Ypma, R. Romero-Garcia, S. J. Price, and J. Suckling. "Graph theory analysis of complex brain networks: new concepts in brain mapping applied to neurosurgery". *Journal of neurosurgery* 124:6, 2016, pp. 1665–1678.

80. A. M. Hasan, H. A. Jalab, F. Meziane, H. Kahtan, and A. S. Al-Ahmad. "Combining deep and handcrafted image features for MRI brain scan classification". *IEEE Access* 7, 2019, pp. 79959–79967.

81. S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann. "On the interpretation of weight vectors of linear models in multivariate neuroimaging". *Neuroimage* 87, 2014, pp. 96–110.

82. D. Haussler. *Convolution kernels on discrete structures*. Technical report. Technical report, Department of Computer Science, University of California, 1999.

83. M. P. van den Heuvel, R. C. Mandl, C. J. Stam, R. S. Kahn, and H. E. H. Pol. "Aberrant frontal and temporal complex network structure in schizophrenia: a graph theoretical analysis". *Journal of Neuroscience* 30:47, 2010, pp. 15915–15926.

84. S. Hido and H. Kashima. "A linear-time graph kernel". In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE. 2009, pp. 179–188.

85. J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. "Cycada: Cycle-consistent adversarial domain adaptation". In: *International conference on machine learning*. PMLR. 2018, pp. 1989–1998.

86. T. Hofmann, B. Schölkopf, and A. J. Smola. "Kernel methods in machine learning". *The annals of statistics*, 2008, pp. 1171–1220.

87. K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators." *Neural networks* 2:5, 1989, pp. 359–366.

88. B. Hu, J. Rao, X. Li, T. Cao, J. Li, D. Majoe, and J. Gutknecht. "Emotion regulating attentional control abnormalities in major depressive disorder: an event-related potential study". *Scientific reports* 7:1, 2017, pp. 1–21.

89. W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec. "Strategies for Pre-training Graph Neural Networks". In: *International Conference on Learning Representations*. 2020.

90. S. A. Huettel, A. W. Song, G. McCarthy, *et al. Functional magnetic resonance imaging*. Vol. 1. Sinauer Associates Sunderland, MA, 2004.

91. M Inglese and M. Bester. "Diffusion imaging in multiple sclerosis: research and clinical implications". *NMR in Biomedicine* 23:7, 2010, pp. 865–872.

92. M. R. Islam, X. Yin, A. Ulhaq, Y. Zhang, H. Wang, N. Anjum, and T. Kron. "A survey of graph based complex brain network analysis using functional and diffusional MRI". *American Journal of Applied Sciences* 14:12, 2018, pp. 1186–1208.

93. S. L. James, D. Abate, K. H. Abate, S. M. Abay, C. Abbafati, N. Abbasi, H. Abbastabar, F. Abd-Allah, J. Abdela, A. Abdelalim, *et al.* "Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017". *The Lancet* 392:10159, 2018, pp. 1789–1858.

94. M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith. "Fsl". *Neuroimage* 62:2, 2012, pp. 782–790.

95. A. M. Kanner. "Is major depression a neurologic disorder with psychiatric symptoms?" *Epilepsy & behavior* 5:5, 2004, pp. 636–644.

96. L. V. Kantorovich. "On the translocation of masses". In: *Dokl. Akad. Nauk. USSR (NS)*. Vol. 37. 1942, pp. 199–201.

97. H. Kashima, K. Tsuda, and A. Inokuchi. "Marginalized kernels between labeled graphs". In: *Proceedings of the 20th international conference on machine learning*. 2003, pp. 321–328.

98. T. Kataoka and A. Inokuchi. "Hadamard Code Graph Kernels for Classifying Graphs." In: *ICPRAM*. 2016, pp. 24–32.

99. K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann. *Benchmark Data Sets for Graph Kernels*. 2016. URL: http://graphkernels.cs.tu-dortmund.de.

100. T. N. Kipf and M. Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations (ICLR)*. 2017.

101. S. Kolouri, Y. Zou, and G. K. Rohde. "Sliced Wasserstein kernels for probability distributions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5258–5267.

102. N. Kriege and P. Mutzel. "Subgraph Matching Kernels for Attributed Graphs". In: *Proceedings of the 29th International Conference on Machine Learning*. 2012, pp. 1015–1022.

103. N. M. Kriege, F. D. Johansson, and C. Morris. "A survey on graph kernels". *Applied Network Science* 5:1, 2020, pp. 1–42.

104. N. M. Kriege, M. Neumann, C. Morris, K. Kersting, and P. Mutzel. "A unifying view of explicit and implicit feature maps of graph kernels". *Data Mining and Knowledge Discovery* 33:6, 2019, pp. 1505–1547.

105. N. M. Kriege, P.-L. Giscard, and R. Wilson. "On valid optimal assignment kernels and applications to graph classification". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1623–1631.

106. A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

107. M. Kuhn, I. Letunic, L. J. Jensen, and P. Bork. "The SIDER database of drugs and side effects". *Nucleic acids research* 44:D1, 2016, pp. D1075–D1079.

108. J. F. Kurtzke. "Rating neurologic impairment in multiple sclerosis: an expanded disability status scale (EDSS)". *Neurology* 33:11, 1983, pp. 1444–1444.

109. G. Landrum *et al.* "RDKit: Open-source cheminformatics", 2006.

110. X. Li and J. Duncan. "BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis". *bioRxiv*, 2020.

111. P. Liu, X. Qiu, and X.-J. Huang. "Adversarial Multi-task Learning for Text Classification". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 1–10.

112. G. Lohmann, J. Stelzer, V. Zuber, T. Buschmann, D. Margulies, A. Bartels, and K. Scheffler. "Task-related edge density (TED)—a new method for revealing dynamic network formation in fMRI data of the human brain". *PLoS One* 11:6, 2016.

113. M. Long, H. Zhu, J. Wang, and M. I. Jordan. "Deep transfer learning with joint adaptation networks". In: *International conference on machine learning*. PMLR. 2017, pp. 2208–2217.

114. M. Long, Y. Cao, J. Wang, and M. Jordan. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

115. G. Loosli, S. Canu, and C. S. Ong. "Learning SVM in Kreĭn spaces". *IEEE transactions on pattern analysis and machine intelligence* 38:6, 2015, pp. 1204–1216.

116. F. D. Lublin, S. C. Reingold, J. A. Cohen, G. R. Cutter, P. S. Sørensen, A. J. Thompson, J. S. Wolinsky, L. J. Balcer, B. Banwell, F. Barkhof, *et al.* "Defining the clinical course of multiple sclerosis: the 2013 revisions". *Neurology* 83:3, 2014, pp. 278–286.

117. X. Lv, Y. Guan, and B. Deng. "Transfer learning based clinical concept extraction on data from multiple sources". *Journal of biomedical informatics* 52, 2014, pp. 55–64.

118. P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. "Extensions of marginalized graph kernels". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004.

119. A. Mahmoudi, S. Takerkart, F. Regragui, D. Boussaoud, and A. Brovelli. "Multivoxel pattern analysis for FMRI data: a review". *Computational and mathematical methods in medicine*, 2012.

120. I. F. Martins, A. L. Teixeira, L. Pinheiro, and A. O. Falcao. "A Bayesian approach to in silico blood-brain barrier penetration modeling". *Journal of chemical information and modeling* 52:6, 2012, pp. 1686–1697.

121. C. McDaniel and S. Quinn. "Developing a Graph Convolution-Based Analysis Pipeline for Multi-Modal Neuroimage Data: An Application to Parkinson's Disease", 2019.

122. F. Mémoli. "Gromov–Wasserstein distances and the metric approach to object matching". *Foundations of computational mathematics* 11:4, 2011, pp. 417–487.

123. F. Mémoli. "Metric structures on datasets: stability and classification of algorithms". In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2011, pp. 1–33.

124. G. Monge. "Mémoire sur la théorie des déblais et des remblais". *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

125. S. Montgomery and M. Åsberg. *A new depression scale designed to be sensitive to change*. Acad. Department of Psychiatry, Guy's Hospital, 1977.

126. C. Morris, N. M. Kriege, K. Kersting, and P. Mutzel. "Faster kernels for graphs with continuous attributes via hashing". In: *16th International Conference on Data Mining (ICDM)*. IEEE. 2016, pp. 1095–1100.

127. S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett. "The Alzheimer's disease neuroimaging initiative". *Neuroimaging Clinics* 15:4, 2005, pp. 869–877.

128. B. J. Nagel and C. D. Kroenke. "The use of magnetic resonance spectroscopy and magnetic resonance imaging in alcohol research". *Alcohol Research & Health* 31:3, 2008, p. 243.

129. N. Navarin, D. V. Tran, and A. Sperduti. "Universal Readout for Graph Convolutional Neural Networks". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–7.

130. M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting. "Propagation kernels: efficient graph kernels from propagated information". *Machine Learning* 102:2, 2016, pp. 209–245.

131. T. E. Nichols. "Multiple testing corrections, nonparametric methods, and random field theory". *Neuroimage* 62:2, 2012, pp. 811–815.

132. R. K. Nielsen, A. N. Holm, and A. Feragen. "Learning from graphs with structural variation". *In NIPS 2017 workshop "Learning on Distributions, Functions, Graphs and Groups"*, 2018.

133. P. A. Novick, O. F. Ortiz, J. Poelman, A. Y. Abdulhay, and V. S. Pande. "SWEETLEAD: an in silico database of approved drugs, regulated chemicals, and herbal isolates for computer-aided drug discovery". *PLoS One* 8:11, 2013.

134. E. E. O'Connor and T. A. Zeffiro. "Why is Clinical fMRI in a Resting State?" *Frontiers in Neurology* 10, 2019, p. 420.

135. D. Oglic and T. Gärtner. "Learning in reproducing kernel Kreın spaces", 2018.

136. S.-i. Ohta. "Barycenters in Alexandrov spaces of curvature bounded below". *Advances in geometry* 12:4, 2012, pp. 571–587.

137. L. Oliveira, C. D. Ladouceur, M. L. Phillips, M. Brammer, and J. Mourao-Miranda. "What does brain response to neutral faces tell us about major depression? Evidence from machine learning and fMRI". *PloS one* 8:4, 2013.

138. C. S. Ong, X. Mary, S. Canu, and A. J. Smola. "Learning with non-positive kernels". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 81.

139. S. Pan and Q. Yang. "A Survey on Transfer Learning". *IEEE Transactions on Knowledge and Data Engineering* 22:10, 2010, pp. 1345–1359.

140. X. Peng, Z. Huang, X. Sun, and K. Saenko. "Domain agnostic learning with disentangled representations". *arXiv preprint arXiv:1904.12347*, 2019.

141. W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Elsevier, 2011.

142. G. Peyré and M. Cuturi. "Computational Optimal Transport: With Applications to Data Science". *Foundations and Trends® in Machine Learning* 11:5-6, 2019, pp. 355–607.

143. D. A. Pizzagalli. "Frontocingulate dysfunction in depression: toward biomarkers of treatment response". *Neuropsychopharmacology* 36:1, 2011, pp. 183–206.

144. F Prados Carrasco, M. J. Cardoso, N. Burgos, C. Wheeler-Kingshott, and S. Ourselin. "NiftyWeb: web based platform for image processing on the cloud". In: International Society for Magnetic Resonance in Medicine (ISMRM). 2016.

145. J. Rabin, G. Peyré, J. Delon, and M. Bernot. "Wasserstein barycenter and its application to texture mixing". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2011, pp. 435–446.

146. M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. "Transfusion: Understanding transfer learning for medical imaging". In: *Advances in neural information processing systems*. 2019, pp. 3347–3357.

147. B. Ramsundar, P. Eastman, P. Walters, and V. Pande. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. O'Reilly Media, 2019.

148. B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. "Massively multitask networks for drug discovery". *arXiv preprint arXiv:1502.02072*, 2015.

149. R. Redlich, N. Opel, D. Grotegerd, K. Dohm, D. Zaremba, C. Bürger, S. Münker, L. Mühlmann, P. Wahl, W. Heindel, *et al.* "Prediction of individual response to electroconvulsive therapy via machine learning on structural magnetic resonance imaging data". *JAMA psychiatry* 73:6, 2016, pp. 557–564.

150. M. Reuter, N. J. Schmansky, H. D. Rosas, and B. Fischl. "Within-subject template estimation for unbiased longitudinal image analysis". *Neuroimage* 61:4, 2012, pp. 1402–1418.

151. P. A. Rinck. *Magnetic resonance in medicine: a critical introduction*. BoD–Books on Demand, 2019.

152. M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich. "To transfer or not to transfer". In: *In NIPS'05 Workshop, Inductive Transfer: 10 Years Later*. 2005.

153. Y. Rubner, C. Tomasi, and L. J. Guibas. "The earth mover's distance as a metric for image retrieval". *International journal of computer vision* 40:2, 2000, pp. 99–121.

154. A. Sartorius, T. Demirakca, A. Böhringer, C. C. von Hohenberg, S. S. Aksay, J. M. Bumb, L. Kranaster, and G. Ende. "Electroconvulsive therapy increases temporal gray matter volume and cortical thickness". *European Neuropsychopharmacology* 26:3, 2016, pp. 506–517.

155. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The Graph Neural Network Model". *IEEE Transactions on Neural Networks* 20:1, 2009, pp. 61–80.

156. M. Scheidegger, A. Henning, M. Walter, H. Boeker, A. Weigand, E. Seifritz, and S. Grimm. "Effects of ketamine on cognition–emotion interaction in the brain". *Neuroimage* 124, 2016, pp. 8–15.

157. D. W. Shattuck and R. M. Leahy. "BrainSuite: an automated cortical surface identification tool". *Medical image analysis* 6:2, 2002, pp. 129–142.

158. J. Shawe-Taylor, N. Cristianini, *et al. Kernel methods for pattern analysis*. Cambridge university press, 2004.

159. N. Shervashidze and K. Borgwardt. "Fast subtree kernels on graphs". In: *Advances in neural information processing systems*. 2009, pp. 1660–1668.

160. N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. "Efficient graphlet kernels for large graph comparison". In: *Artificial Intelligence and Statistics*. 2009, pp. 488–495.

161. N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt. "Weisfeiler-Lehman graph kernels". *Journal of Machine Learning Research* 12, 2011, pp. 2539–2561.

162. G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis. "GraKeL: A Graph Kernel Library in Python." *Journal of Machine Learning Research* 21:54, 2020, pp. 1–5.

163. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.* "Mastering the game of go without human knowledge". *Nature* 550:7676, 2017, pp. 354–359.

164. O. Sporns. "The human connectome: a complex network". *Annals of the New York Academy of Sciences* 1224:1, 2011, pp. 109–125.

165. S. S. Stevens *et al.* "On the theory of scales of measurement", 1946.

166. G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny. "Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches". *Journal of chemical information and modeling* 56:10, 2016, pp. 1936–1949.

167. M. Sugiyama and K. Borgwardt. "Halting in random walk kernels". In: *Advances in neural information processing systems*. 2015, pp. 1639–1647.

168. M. Sugiyama, M. E. Ghisu, F. Llinares-López, and K. Borgwardt. "graphkernels: R and Python packages for graph comparison". *Bioinformatics* 34:3, 2018, pp. 530–532.

169. J. J. Sutherland, L. A. O'brien, and D. F. Weaver. "Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships". *Journal of chemical information and computer sciences* 43:6, 2003, pp. 1906–1915.

170. X. Tang, Y. Li, Y. Sun, H. Yao, P. Mitra, and S. Wang. "Transferring Robustness for Graph Neural Network Against Poisoning Attacks". In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. 2020, pp. 600–608.

171. B. Thirion, P. Pinel, S. Mériaux, A. Roche, S. Dehaene, and J.-B. Poline. "Analysis of a large fMRI cohort: Statistical and methodological issues for group analyses". *Neuroimage* 35:1, 2007, pp. 105–120.

172. M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt. "Wasserstein weisfeiler-lehman graph kernels". In: *Advances in Neural Information Processing Systems*. 2019, pp. 6439–6449.

173. A. Tousignant, P. Lemaître, D. Precup, D. L. Arnold, and T. Arbel. "Prediction of disease progression in multiple sclerosis patients using deep learning analysis of MRI data". In: *International Conference on Medical Imaging with Deep Learning*. 2019, pp. 483–492.

174. K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. "Fréchet means for distributions of persistence diagrams". *Discrete & Computational Geometry* 52:1, 2014, pp. 44–70.

175. E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial Discriminative Domain Adaptation". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2962–2971.

176. *UK Biobank*. URL: \url{https://www.ukbiobank.ac.uk/}.

177. L. Van Diermen, S. Van Den Ameele, A. M. Kamperman, B. C. Sabbe, T. Vermeulen, D. Schrijvers, and T. K. Birkenhäger. "Prediction of electroconvulsive therapy response and remission in major depression: meta-analysis". *The British Journal of Psychiatry* 212:2, 2018, pp. 71–80.

178. G. Varoquaux. "Cross-validation failure: small sample sizes lead to large error bars". *Neuroimage* 180, 2018, pp. 68–77.

179. G. Varoquaux, P. R. Raamana, D. A. Engemann, A. Hoyos-Idrobo, Y. Schwartz, and B. Thirion. "Assessing and tuning brain decoders: cross-validation, caveats, and guidelines". *NeuroImage* 145, 2017, pp. 166–179.

180. S. Vega-Pons and P. Avesani. "Brain decoding via graph kernels". In: *2013 International Workshop on Pattern Recognition in Neuroimaging*. IEEE. 2013, pp. 136–139.

181. S. Vega-Pons, P. Avesani, M. Andric, and U. Hasson. "Classification of inter-subject fMRI data based on graph kernels". In: *2014 International Workshop on Pattern Recognition in Neuroimaging*. IEEE. 2014, pp. 1–4.

182. J.-P. Vert. "The optimal assignment kernel is not positive definite". *arXiv preprint arXiv:0801.4061*, 2008.

183. C. Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

184. S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. "Graph kernels". *Journal of Machine Learning Research* 11, 2010, pp. 1201–1242.

185. M. L. Vo, M. Conrad, L. Kuchinke, K. Urton, M. J. Hofmann, and A. M. Jacobs. "The Berlin affective word list reloaded (BAWL-R)". *Behavior research methods* 41:2, 2009, pp. 534–538.

186. U. Von Luxburg. "A tutorial on spectral clustering". *Statistics and computing* 17:4, 2007, pp. 395–416.

187. J. A. van Waarde, L. J. van Oudheusden, O. B. Heslinga, B. Verwey, R. C. van der Mast, and E. Giltay. "Patient, treatment, and anatomical predictors of outcome in electroconvulsive therapy: a prospective study". *The journal of ECT* 29:2, 2013, pp. 113–121.

188. T. D. Wager and E. E. Smith. "Neuroimaging studies of working memory". *Cognitive, Affective, & Behavioral Neuroscience* 3:4, 2003, pp. 255–274.

189. X.-L. Wang, M.-Y. Du, T.-L. Chen, Z.-Q. Chen, X.-Q. Huang, Y. Luo, Y.-J. Zhao, P. Kumar, and Q.-Y. Gong. "Neural correlates during working memory processing in major depressive disorder". *Progress in Neuro-Psychopharmacology and Biological Psychiatry* 56, 2015, pp. 101–108.

190. S. Warshall. "A theorem on boolean matrices". *Journal of the ACM (JACM)* 9:1, 1962, pp. 11–12.

191. C.-Y. Wee, C. Liu, A. Lee, J. S. Poh, H. Ji, A. Qiu, A. D. N. Initiative, *et al.* "Cortical graph neural network for AD and MCI diagnosis and transfer learning across populations". *NeuroImage: Clinical* 23, 2019, p. 101929.

192. B. Weisfeiler and A. A. Lehman. "A reduction of a graph to a canonical form and and algebra arising during this reduction". *Nauchno-Technicheskaya Informatsia, Ser. 2* 9, 1968.

193. S. Whitfield-Gabrieli and A. Nieto-Castanon. "Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks". *Brain connectivity* 2:3, 2012, pp. 125–141.

194. M. Wu, S. Pan, X. Zhu, C. Zhou, and L. Pan. "Domain-adversarial graph neural networks for text classification". In: *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2019, pp. 648–657.

195. M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu. "Unsupervised Domain Adaptive Graph Convolutional Networks". In: *Proceedings of The Web Conference 2020*. 2020, pp. 1457–1467.

196. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. "A Comprehensive Survey on Graph Neural Networks". *IEEE Transactions on Neural Networks and Learning Systems*, 2020, pp. 1–21.

197. Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. "MoleculeNet: a benchmark for molecular machine learning". *Chemical Science* 9:2, 2018, pp. 513–530.

198. K. Xu, W. Hu, J. Leskovec, and S. Jegelka. "How Powerful are Graph Neural Networks?" In: *International Conference on Learning Representations*. 2019.

199. P. Yanardag and S. Vishwanathan. "A structural smoothing framework for robust graph comparison". In: *Advances in neural information processing systems*. 2015, pp. 2134–2142.

200. P. Yanardag and S. Vishwanathan. "Deep graph kernels". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015, pp. 1365–1374.

201. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

202. V. Zamoscik, S. Huffziger, U. Ebner-Priemer, C. Kuehner, and P. Kirsch. "Increased involvement of the parahippocampal gyri in a sad mood predicts future depressive symptoms". *Social cognitive and affective neuroscience* 9:12, 2014, pp. 2034–2040.

203. L.-L. Zeng, H. Shen, L. Liu, L. Wang, B. Li, P. Fang, Z. Zhou, Y. Li, and D. Hu. "Identifying major depression using whole-brain functional connectivity: a multivariate pattern analysis". *Brain* 135:5, 2012, pp. 1498–1507.

204. Y. Zhou, X. Mei, W. Li, and J. Huang. "Classification of resting-state fMRI datasets based on graph kernels". In: *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE. 2017, pp. 665–669.

205. D. Zügner and S. Günnemann. "Certifiable robustness and robust training for graph convolutional networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 246–256.