





UNIVERSITAT  
ROVIRA i VIRGILI



UNIVERSITAT DE  
BARCELONA



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# PREDICTING BRAIN CONNECTIVITY MAPPING USING RADIOMICS FEATURES IN ANATOMICAL MRI

LEVENTE ZSOLT NAGY

**Thesis supervisor**

ALFREDO VELLIDO ALCACENA (Department of Computer Science)

**Thesis co-supervisor**

ESTELA CAMARA MANCHA

**Degree**

Master's Degree in Artificial Intelligence

**Master's thesis**

**School of Engineering**

Universitat Rovira i Virgili (URV)

**Faculty of Mathematics**

Universitat de Barcelona (UB)

**Barcelona School of Informatics (FIB)**

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

29/01/2025

## Abstract

Anatomical magnetic resonance imaging (MRI), including T1 and T2 weighted images, is routinely used to visualize anatomical structures and detect pathological changes, as these imaging techniques provide complementary contrasts that highlight distinct tissue properties and abnormalities. Diffusion tensor imaging (DTI), a more specialized MRI technique, enables the mapping of white matter and the study of structural connectivity through metrics like fractional anisotropy (FA) and mean diffusivity (MD). However, DTI acquisition is time intensive and often excluded from standard clinical protocols.

This study introduces a method to synthesize FA and MD images from widely available T1 and T2 weighted anatomical scans, reducing dependence on resource intensive DTI acquisition.

Radiomics, a rapidly evolving field, focuses on extracting quantitative features from medical imaging, to identify biomarkers associated with clinical labels. It is a well established tool in oncology, particularly for tumor segmentation. Fully convolutional neural networks (FCNNs) are also employed for similar tasks, such as deriving clinical labels for tumor segmentation. However, their reliance on computationally intensive 3D convolutions often necessitates the use of 2D slices from 3D volumes, limiting their feasibility.

We present a hybrid approach that combines the strengths of radiomics and neural networks. Specifically, we use a feedforward neural network (FNN) as a classification or regression head applied to radiomic features, preserving 3D spatial information during feature extraction while reducing the computational demands of 3D convolutions. This neural network head, akin to fully connected layers in traditional convolutional neural networks (CNNs), offers significant flexibility.

Our experiments are limited to the basal ganglia, a crucial region involved in brain structure and function that is significantly impacted by neurodegeneration in Huntington's disease (HD). The inclusion of HD patients allows for assessing the method's reliability and robustness. Additionally, we explore the potential of using the T1/T2 ratio as an input image, which has been proposed in recent studies, as a proxy for the myelin content of the brain. Since myelin plays a role in how DTI functions, the T1/T2 ratio may enhance model performance, effectively bridging anatomical MRI and DTI.

The reported results of the experiments indicate strong performance, with Pearson correlations of 0.85 for FA and 0.95 for MD predictions. While the correlation decreases significantly for FA and moderately for MD in HD patients, the model's performance remains consistent when both healthy controls and patients are analyzed together. These findings underscore the promise of our hybrid approach for synthesizing structural connectivity images and improving accessibility to diffusion metrics.

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
1.1	Background . . . . .	10
1.1.1	Magnetic Resonance Imaging . . . . .	10
1.1.2	Radiomics . . . . .	11
1.1.3	Machine Learning . . . . .	11
1.1.4	Basal Ganglia and Huntington’s Disease . . . . .	11
1.2	State of the Art . . . . .	12
1.2.1	Myelin Proxy . . . . .	12
1.2.2	Machine Learning . . . . .	12
1.2.3	Radiomics . . . . .	12
1.3	Motivation . . . . .	12
1.4	Objectives . . . . .	13
1.5	Previous Work . . . . .	13
<b>2</b>	<b>Data and Evaluation Workflow</b>	<b>15</b>
2.1	Architecture . . . . .	15
2.2	Data Representation . . . . .	15
2.3	Preprocessing . . . . .	16
2.3.1	Raw Data . . . . .	16
2.3.2	Quality Control . . . . .	18
2.3.3	Radiomics Features . . . . .	19
2.3.4	Coordinates . . . . .	20
2.3.5	Data Augmentation . . . . .	20
2.3.6	Scaling and Normalization . . . . .	20
2.3.7	Data Balancing . . . . .	22
2.3.8	Clinical Data . . . . .	23
2.3.9	Relative Connectivity . . . . .	23
2.4	Evaluation . . . . .	24
2.4.1	Train, Validation and Test Splits . . . . .	24
2.4.2	Accuracy and Pearson Correlation . . . . .	24
<b>3</b>	<b>Experiments</b>	<b>26</b>
3.1	Subcortical Segmentation . . . . .	26
3.2	Methodology . . . . .	28
3.2.1	Native or Normalized Space . . . . .	28
3.2.2	T1 or T1/T2 Imaging . . . . .	28
3.2.3	Further Hyperparameters . . . . .	28
3.2.4	Missing Records . . . . .	29
3.2.5	Architecture Tuning . . . . .	30
3.3	Diffusion Fractional Anisotropy Regression . . . . .	30

3.4	Mean Diffusivity Regression . . . . .	33
3.5	Relative Connectivity Segmentation . . . . .	36
3.5.1	Exhaustive Sequential Backwards Feature Selection . . . . .	41
3.5.2	Streamline Regression . . . . .	42
<b>4</b>	<b>Sustainability</b>	<b>43</b>
4.1	Environmental Aspect . . . . .	43
4.1.1	Developement . . . . .	43
4.1.2	Production . . . . .	44
4.2	Economic Aspect . . . . .	44
4.2.1	Cost . . . . .	44
4.2.2	Return . . . . .	45
4.3	Social Aspect . . . . .	46
4.3.1	Inclusivity . . . . .	46
4.4	Risks . . . . .	46
<b>5</b>	<b>Conclusions</b>	<b>47</b>
5.1	Future Improvements . . . . .	48
<b>Sources of Information</b>		<b>50</b>
<b>A</b>	<b>Previous Work</b>	<b>53</b>
A.1	Participants . . . . .	53
A.2	Clinical Evaluation . . . . .	53
A.3	MRI Data Acquisition . . . . .	53
A.4	Tractography . . . . .	54
A.5	Connectivity Calculation . . . . .	54
<b>B</b>	<b>Software Design</b>	<b>55</b>
B.1	Raw Data . . . . .	55
B.2	Common Functions . . . . .	57
B.3	Preprocessed Data . . . . .	58
B.4	Data Generator . . . . .	60
<b>C</b>	<b>Software Implementation</b>	<b>63</b>
C.1	Interpolation . . . . .	63
C.2	Multithreading . . . . .	63
C.3	Warp Pre-Computing . . . . .	64
C.4	Data Normalization . . . . .	64
<b>D</b>	<b>Additional Figures</b>	<b>65</b>
<b>E</b>	<b>Additional Tables</b>	<b>69</b>
<b>F</b>	<b>Source Code</b>	<b>86</b>

# List of Notations & Abbreviations

<b>MRI</b>	Magnetic Resonance Imaging .....	10
<b>DTI</b>	Diffusion Tensor Imaging .....	10
<b>CNS</b>	central nervous system .....	10
<b>FA</b>	fractional anisotropy .....	10
<b>MD</b>	mean diffusivity .....	10
<b>HD</b>	Huntington's disease .....	11
<b>FNN</b>	feedforward neural network .....	11
<b>SVM</b>	support vector machine .....	11
<b>MWF</b>	myelin water fraction .....	12
<b>NN</b>	neural network .....	12
<b>FCNN</b>	fully convolutional neural network .....	12
<b>CNN</b>	convolutional neural network .....	13
<b>ROI</b>	region of interest .....	13
<b>FMRIB</b>	functional magnetic resonance imaging of the brain	
<b>FNIRT</b>	FMRIB's nonlinear image registration tool .....	14
<b>NIfTI</b>	neuroimaging informatics technology initiative .....	16
<b>GLCM</b>	gray level co-occurrence matrix .....	19
<b>GLSZM</b>	gray level size zone matrix .....	19
<b>GLRLM</b>	gray level run length matrix .....	19
<b>NGTDM</b>	neighbouring gray tone difference matrix .....	19
<b>GLDM</b>	gray level dependence matrix .....	19
<b>CAP</b>	CAG Age Product .....	23
<b>cUHDRS</b>	composite Unified Huntington's Disease Rating Scale .....	23
<b>FSL</b>	FMRIB software library .....	54
<b>FDT</b>	FMRIB's diffusion toolbox .....	54
<b>UML</b>	unified modeling language .....	56

# List of Figures

1.1	Physical Barrier Restricting Diffusion in DTI, adopted from [6] . . . . .	10
2.1	Architecture Overview . . . . .	15
2.2	Caudate Subcortical Region Overlapping in Different Spaces . . . . .	17
2.3	Histogram: Firstorder Energy . . . . .	20
2.4	Histogram: GLDM Small Dependence High Gray Level Emphasis . . . . .	21
2.5	Histogram: NGTDM Busyness . . . . .	21
2.6	Partial Data Balancing: Subcortical . . . . .	22
2.7	Partial Data Balancing: Relative Connectivity . . . . .	22
2.8	Partial Data Balancing: Diffusion FA . . . . .	22
2.9	Partial Data Balancing: Diffusion MD . . . . .	23
2.10	Distribution of Records in Relation to Datapoints . . . . .	24
3.1	Training Curve: Diffusion Fractional Anisotropy Best Performing Model . . . . .	31
3.2	Train Predictions: Diffusion Fractional Anisotropy Best Performing Model . . . . .	32
3.3	Validation Predictions: Diffusion Fractional Anisotropy Best Performing Model . . . . .	32
3.4	Test Predictions: Diffusion Fractional Anisotropy Best Performing Model . . . . .	33
3.5	Training Curve: Mean Diffusivity Best Performing Model . . . . .	34
3.6	Train Predictions: Mean Diffusivity Best Performing Model . . . . .	34
3.7	Validation Predictions: Mean Diffusivity Best Performing Model . . . . .	35
3.8	Test Predictions: Mean Diffusivity Best Performing Model . . . . .	35
3.9	Confusion Matrices (Precision): Relative Connectivity Best Performing Model . . . . .	37
3.10	Confusion Matrices (Recall): Relative Connectivity Best Performing Model . . . . .	38
3.11	Training Curve: Relative Connectivity Best Performing Model . . . . .	38
3.12	Basal Ganglia (ROI) & Cortical Targets . . . . .	39
3.13	Train Predictions: Relative Connectivity Best Performing Model . . . . .	39
3.14	Validation Predictions: Relative Connectivity Best Performing Model . . . . .	40
3.15	Test Predictions: Relative Connectivity Best Performing Model . . . . .	40
B.1	Files: Raw . . . . .	55
B.2	Class Diagram: Data Types . . . . .	56
B.3	Class Diagram: Preprocessing . . . . .	57
B.4	Class Diagram: Common . . . . .	58
B.5	Files: Preprocessed . . . . .	59
B.6	Class Diagram: Experimentation . . . . .	60
C.1	Index Array Mapping . . . . .	64
D.1	Loss of Detail in GLDM Small Dependence High Gray Level Emphasis . . . . .	65

D.2	Loss of Detail in NGTDM Busyness . . . . .	66
D.3	Training Curve: Subcortical Segmentation Best Performing Model . . . . .	67
D.4	Train Predictions: Subcortical Segmentation Best Performing Model . . . . .	67
D.5	Validation Predictions: Subcortical Segmentation Best Performing Model . . . . .	68
D.6	Test Predictions: Subcortical Segmentation Best Performing Model . . . . .	68

# List of Tables

2.1	Raw Data . . . . .	16
2.2	Radiomic Feature Types . . . . .	19
3.1	Constant Hyperparameters . . . . .	26
3.2	Initial Hyperparameters . . . . .	26
3.3	Hyperparameter Tuning: Subcortical Experiments . . . . .	27
3.4	Missing Records . . . . .	29
3.5	Cortical Targets Color Coding . . . . .	39
3.6	Feature Selection: Excluded Features . . . . .	42
4.1	Sustainability: Power Consumptions and CO <sub>2</sub> Emissions . . . . .	44
4.2	Project Cost Breakdown . . . . .	45
B.1	Data Generator Properties . . . . .	62
D.1	Subcortical Color Coding . . . . .	67
E.1	Voxel Based Radiomic Features . . . . .	70
E.2	Shape Based Radiomic Features . . . . .	70
E.3	Hyperparameter Tuning: Diffusion Fractional Anisotropy - Native T1 . . . . .	71
E.4	Hyperparameter Tuning: Diffusion Fractional Anisotropy - Native T1/T2 . . . . .	72
E.5	Hyperparameter Tuning: Diffusion Fractional Anisotropy - Normalized T1 . . . . .	73
E.6	Hyperparameter Tuning: Diffusion Fractional Anisotropy - Normalized T1/T2 . . . . .	74
E.7	Architecture Tuning: Diffusion Fractional Anisotropy . . . . .	75
E.8	Hyperparameter Tuning: Mean Diffusivity - Native T1 . . . . .	76
E.9	Hyperparameter Tuning: Mean Diffusivity - Native T1/T2 . . . . .	77
E.10	Hyperparameter Tuning: Mean Diffusivity - Normalized T1 . . . . .	78
E.11	Hyperparameter Tuning: Mean Diffusivity - Normalized T1/T2 . . . . .	79
E.12	Architecture Tuning: Mean Diffusivity . . . . .	80
E.13	Hyperparameter Tuning: Relative Connectivity - Native T1 . . . . .	81
E.14	Hyperparameter Tuning: Relative Connectivity - Native T1/T2 . . . . .	82
E.15	Hyperparameter Tuning: Relative Connectivity - Normalized T1 . . . . .	83
E.16	Hyperparameter Tuning: Relative Connectivity - Normalized T1/T2 . . . . .	84
E.17	Architecture Tuning: Relative Connectivity . . . . .	85

# 1 Introduction

## 1.1 Background

### 1.1.1 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) methods, such as T1 and T2 weighted imaging, measure the relaxation of a tissue's net magnetization vector. [1] These volumetric imaging methods are useful for distinguishing between different tissue types.

T1 relaxation time, also known as longitudinal relaxation time, refers to the time it takes for the protons to realign with the magnetic field after the radiofrequency pulse is turned off. Tissues with high-fat content, such as white matter in the brain, have shorter T1 relaxation times, which is why they appear bright on T1 weighted images. [2]

T2 relaxation time, also known as transverse relaxation time, measures how long it takes for protons to lose phase coherence among the spins perpendicular to the magnetic field. This is influenced by the interaction of spins within the tissue. Tissues with high water content, like cerebrospinal fluid, have longer T2 relaxation times, appearing bright on T2 weighted images. [2]

Neurons, the electrically excitable cells of the brain, process and transmit information via electro-chemical signaling. [3] The main function of the white matter is to connect neurons from different brain regions to each other by myelinated nerve fibers. [4] The myelin sheath is made of cells wrapped around the nerve axon. [5]

Diffusion Tensor Imaging (DTI) is a specialized MRI technique that measures the rate and direction of movement of water molecules within tissues. In the central nervous system (CNS), axonal tracts and myelin present physical barriers that impose directionality or anisotropy on water diffusion (Figure 1.1). Applying magnetic field gradients allows the measurement of the rate and direction of the movement of water molecules in the direction of the magnetic field. Using the basis of diffusion, we can construct a 3D directional architecture of axon fibers and myelin in the CNS. [6]

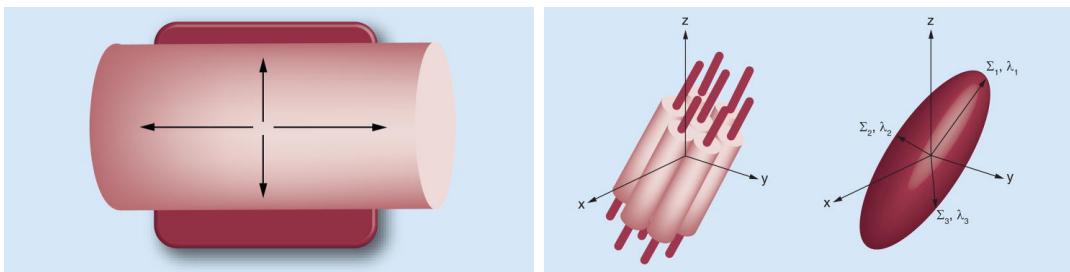


Figure 1.1: Physical Barrier Restricting Diffusion in DTI, adopted from [6]

Certain images extracted from a DTI image, like fractional anisotropy (FA) and mean diffusivity (MD) are invaluable for the characterization, research and diagnosis of neurodegenerative diseases. The FA scalar measures the fraction of the diffusion, where an isotropic medium takes a value of 0 and a cylindrically symmetric anisotropic medium takes a value of 1. The MD scalar measures how freely water molecules move in tissue, where greater tissue density means lower MD. [7]

Furthermore, a more complicated algorithm can be employed to extract information regarding the connectivity of brain regions, specifically using tractography methods to model the trajectories of white matter pathways. Tractography algorithms, such as probabilistic tractography, can simulate numerous potential pathways (tracts) originating from a seed region. These tracts can be traced to determine their likelihood of reaching specific target regions. Each voxel within the seed region can be assigned a connectivity value based on the proportion of simulated tracts that successfully reach a given target region, providing a probabilistic measure of connectivity. Once these connectivity values are obtained, the seed region can be either segmented, or parcellated, into distinct subregions that are preferentially connected to different targets. This connectivity based parcellation enables a detailed examination of the structural organization of brain regions and their relationship with target areas, contributing valuable insights into both healthy and pathological brain networks. [8] [9]

However, due to DTI not being part of many clinical protocols, and it takes a relatively long time to acquire, it is not widely available in actual practice.

### 1.1.2 Radiomics

Although the term is not strictly defined, radiomics generally refers to the extraction of quantitative, and ideally reproducible, information from diagnostic images, including complex patterns that are difficult to recognize or quantify by the human eye. [10] Radiomic features capture tissue characteristics such as heterogeneity, shape, texture, intensity, spatial relationships, edge sharpness, contrast and much more. [11] Radiomic features can be extracted using voxel-based and non-voxel-based approaches. A voxel is the atomic element of a volumetric image, such as an MRI record (analogous to a pixel in a 2D image). In the voxel-based method, features are derived from each voxel and its surrounding area using a kernel-based approach similar to convolution. Conversely, the non-voxel-based method extracts features from any arbitrary mask used as a kernel, which could encompass specific regions or even the entire brain.

### 1.1.3 Machine Learning

Machine learning has transformed the approach to data processing and analysis, enabling the development of models that can identify patterns and make predictions based on previous data [12]. Key techniques within this field include feedforward neural network (FNN) and support vector machine (SVM), both of which are essential tools for solving classification and regression problems [13] [14]. FNNs operate by passing input data through multiple layers of interconnected nodes, progressively transforming the data to extract meaningful insights. On the other hand, SVMs focus on identifying the optimal hyperplane that best separates data points in high-dimensional spaces, balancing simplicity with effectiveness, particularly in situations where data is limited. While many other methods exist within machine learning, FNNs and SVMs remain some of the most widely adopted due to their flexibility and versatility.

### 1.1.4 Basal Ganglia and Huntington's Disease

Basal ganglia are a part of the human brain which is a group of subcortical nuclei responsible primarily for motor control, as well as other roles such as motor learning, executive functions, behaviors, and emotions. [15] Huntington's disease (HD) is a disorder that causes the progressive

degeneration of the basal nuclei. [16] This manifests as both structural connectivity changes and anatomical volume changes. [17]

## 1.2 State of the Art

### 1.2.1 Myelin Proxy

Research has shown that the ratio of commonly acquired T1 and T2 weighted images can serve as a proxy for myelin content. This suggests that structural connectivity information can be indirectly derived from these anatomical scans. For instance, one study [18] successfully mapped myelin content in the brain using the T1/T2 ratio. Similarly, another study [5] utilized this ratio to delineate cortical areas of the brain based on myelin distribution. Additionally, this study [19] identified direct correlations between the myelin water fraction (MWF) and the T1/T2 ratio.

### 1.2.2 Machine Learning

Machine learning has also become increasingly prominent in medical imaging, particularly in tumor segmentation. Architectures like U-Net, a type of fully convolutional neural network (FCNN), have proven highly effective in this domain. Numerous studies [20] [21] [22] [23] [24] employ solutions built on similar architectures. However, most of these methods operate on 2D slices of 3D volumes, as 2D convolutional backbones are better explored and computationally more efficient, avoiding the exponential increase in resource demands associated with higher dimensions.

Generalized machine learning approaches, including FNNs, have proven particularly effective for managing large datasets [25]. Alternative techniques, such as SVMs, have also been explored to tackle their scaling issues for larger datasets [26] [27]. Nonetheless, FNNs remain the preferred solution due to their adaptability and superior performance for large, high-dimensional data.

### 1.2.3 Radiomics

Radiomics can also facilitate the identification of image-derived biomarkers that correlate with specific clinical labels, such as tumor segmentation. [28] [29] [30] A key advantage of radiomics is that unlike neural network (NN)-based methods, radiomics can efficiently operate on 3D volumes without the significantly higher computational demands that often render 3D neural networks less feasible in practice.

## 1.3 Motivation

The state-of-the-art studies highlight an intriguing possibility: can structural connectivity images be synthesized directly from anatomical T1 and T2 weighted images using machine learning? Such a development could bypass the time- and resource-intensive processes associated with DTI acquisition and post-processing, such as tractography. Furthermore, it would allow the synthesis of these less common images from historical datasets that often only include anatomical scans, greatly enhancing their utility in retrospective analyses.

FCNN and radiomics based approaches have demonstrated exceptional accuracy in tasks like tumor segmentation, showcasing their potential for high-precision medical imaging applications.

However, surprisingly few studies have explored synthesizing DTI related images from T1 and T2 data. Moreover, the integration of radiomics with NN-based methods remains underexplored, despite the complementary nature of these techniques. Radiomic features, extracted voxel-based from 3D volumes, are conceptually parallel to the feature extraction role of convolutional neural network (CNN) backbones but offer unique advantages in handling 3D data.

By combining radiomics with NNs, a hybrid approach could capitalize on the strengths of both paradigms. Specifically, appending a NN head for classification or regression to voxel-based radiomic features would enable the use of real 3D information without incurring the significant computational overhead associated with 3D CNNs. This hybrid strategy could pave the way for efficient and accurate synthesis of structural connectivity images, leveraging the elasticity of NNs while retaining the richness of 3D volumetric features.

## 1.4 Objectives

The primary objective of this research is to integrate radiomics-based feature extraction with NNs to predict FA, MD and relative connectivity from T1 and T2 weighted images. The study aims to explore voxel-based features, potentially in combination with non-voxel-based features extracted from several cortical areas. Since the feature extraction process relies on non-NN techniques, the classification or regression head will utilize a FNN, akin to the fully connected layers of a CNN following the flattening of convolutional layers.

Additionally, this work will assess the robustness of the proposed method by conducting experiments on datasets containing healthy control participants and those affected by HD neurodegeneration, through separation or combination of the two groups.

The scope of this project has been intentionally narrowed to ensure its feasibility within the timeframe of the thesis. Specifically, the research focuses on the basal ganglia as the primary region of interest (ROI). This decision allows for a concentrated examination of a well-defined brain area closely associated with structural and functional connectivity, while managing the computational and methodological demands of the study.

Moreover, the study is designed as a simplified proof of concept, utilizing a limited dataset. This constraint reflects the exploratory nature of the work, aiming to establish the feasibility of synthesizing structural connectivity images from T1 and T2 weighted scans using the proposed methodology. The results are therefore not intended to represent a generalized or universally applicable solution but rather to demonstrate the potential of this approach for further, more expansive investigations.

It is important to note that this research does not aim to cover the full spectrum of structural connectivity imaging across the brain or account for variations introduced by diverse pathological or demographic factors. Instead, the findings are expected to provide a foundation for subsequent studies that could explore broader generalizability, larger datasets and more comprehensive analyses of multiple brain regions. These limitations should be considered when interpreting the results and their potential applications.

## 1.5 Previous Work

Detailed data acquisition, pre- and post-processing steps of the previous work done at Hospital de Bellvitge (L'Hospitalet de Llobregat, Barcelona, Spain) are in Appendix A. In summary, Hospital

de Bellvitge provided an extensive dataset comprising anatomical MRI and DTI records. This includes data from 32 control subjects and 38 HD patients, featuring T1-weighted MRI images with isotropic voxels of 1 mm resolution, T2-weighted images with anisotropic 1x1x2 mm resolution, and DTI FA and MD images with isotropic voxels of 2 mm resolution. Furthermore, this dataset also contains the mask for the basal ganglia, which will also be referenced as the ROI; masks for the 7 cortical regions of the brain, which will also be referenced as the target regions: Limbic, Executive, Rostral-Motor, Caudal-Motor, Parietal, Occipital and Temporal; streamline images and relative connectivity maps from the results of tractography; subcortical segmentation of the basal ganglia for the Caudate, Putamen and Accumbens; and FMRIB’s nonlinear image registration tool (FNIRT) warp fields for converting the records into normalized space.

Furthermore, for both the ROI and cortical targets, the dataset distinguishes between the right and left hemispheres of the brain. Thus, there are actually 2 ROIs and, consequently,  $2 \cdot 7 = 14$  target regions.

## 2 Data and Evaluation Workflow

The proposed analytical design integrates voxel-based and non-voxel-based radiomic feature extraction from T1 and T1/T2 images with a NN model to predict structural connectivity metrics, including FA, MD and relative connectivity values.

### 2.1 Architecture

The core architecture, as schematically represented in Figure 2.1, consists of two primary components:

- **Feature Extraction:** This step involves generating voxel-based and non-voxel-based feature vectors from the T1 and T1/T2 images.
- **FNN:** A regression or classification head that processes the extracted feature vectors to predict target outputs, such as FA, MD or relative connectivity values.

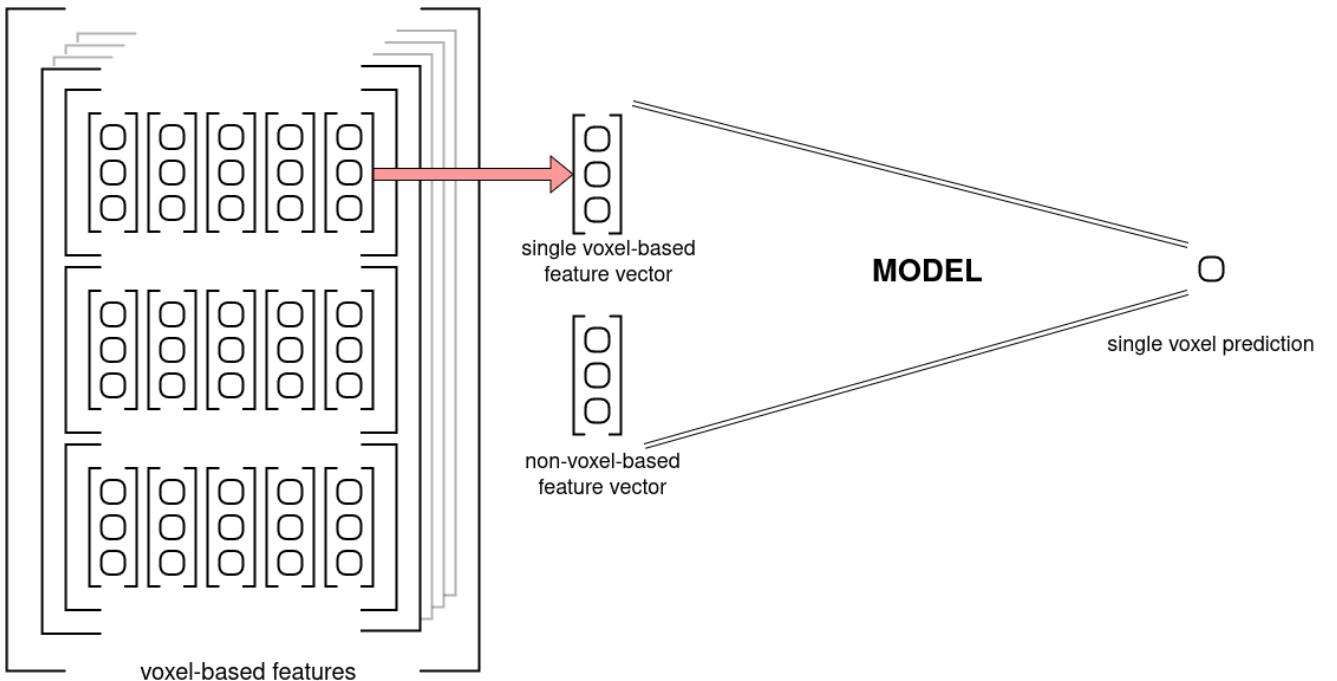


Figure 2.1: Architecture Overview

### 2.2 Data Representation

A crucial distinction exists between volumetric spatial data (such as T1 and T1/T2 images or their corresponding voxel-based features) and individual feature vectors. While the model operates at

the feature vector level, lacking global context, certain evaluation processes are performed at the record level.

**This report will refer to the control/patient spatial data as 'Record' ("voxel-based features" in Figure 2.1) and to individual feature vectors as 'Datapoint' ("single voxel-based feature vector" and "non-voxel-based feature vector" in Figure 2.1).**

## 2.3 Preprocessing

### 2.3.1 Raw Data

All provided records are in the neuroimaging informatics technology initiative (NIfTI) format, which need to be parsed. This format stores the raw output of the MRI record (an example of which is shown in Table 2.1), and additionally an affine transformation matrix used for aligning different spaces.

Data	Shape	Range	Type	Space	Reference
DTI	(118, 118, 60, 74)	[0, 4096]	uint	diffusion	diffusion
Diffusion FA	(118, 118, 60)	[0, 1]	float	diffusion	diffusion_fa
Diffusion MD	(118, 118, 60)	[0, 0.01]	float	diffusion	diffusion_md
T1	(208, 256, 256)	[0, 1000]	float	t1	t1
T1/T2	(208, 256, 256)	[0, 1]	float	d_aligned	t1t2
Cortical Targets	(118, 118, 60, 14)	{0, 1}	bool	diffusion	targets
Relative Connectivity	(118, 118, 60, 14)	[0, 1]	float	diffusion	connectivity
Streamline Image	(118, 118, 60, 14)	[0, 5000]	uint	diffusion	streamline
ROI Mask (Basal Ganglia)	(118, 118, 60, 2)	{0, 1}	bool	diffusion	mask_basal & roi
Brain Mask	(208, 256, 256)	{0, 1}	bool	t1	mask_brain
Basal Ganglia Segmentation	(208, 256, 256)	[0, 58]	uint	t1	basal_seg

Table 2.1: Raw Data

#### 2.3.1.a Brain Mask

The provided dataset did not apply the brain masks for the T1 images out of the box so it can be done with a simple element-wise multiplication of the T1 image and T1 mask.

#### 2.3.1.b Registration

The process of aligning different records into the same native space is called "registration". The provided dataset comes with 2 (3) different spaces, earlier referenced to as t1 and diffusion (and d\_aligned). Most of the data are in diffusion space, thus it is logical to register the rest into the same space. After manual inspection, only 15 records required registration, out of which only 3 required a tiny translation, while the remaining 12 needed a complete affine registration.

The image T1/T2 is the odd one out, as it is inherently in a different space from diffusion (due to them being different resolution), but they are aligned into diffusion space. Although they are practically already registered, this has to be taken into account later on.

### 2.3.1.c Normalization

The process of warping each brain into a common space is called "normalization". Applying the FNIRT warp fields is more or less straightforward, as two warp fields are provided, one for the diffusion space and one for the T1 space. Note that this process inherently contains the benefits of registration, as it is warping the different images into a common brain shape and space. This also paves the direction of future experiments, as it opens the door to working in either native or normalized space.

The only encountered obstacle was with the T1/T2 image. As it is aligned in diffusion space, but FNIRT convention ignores the affine transformation of the NIfTI format, thus making its registration useless as the raw data of the t1t2 has nothing to do with the raw diffusion data (due to them being different resolution). The solution is to apply an affine matrix to t1t2's raw data which transforms it into t1's raw data space, after which the t1's FNIRT warp field can be applied to the t1t2 image. This affine transformation matrix can be easily calculated from the already given matrices. Let  $A$  denote T1/T2's affine matrix and  $B$  denote T1's affine matrix (after registration), thus the matrix which transforms the T1/T2 into T1 space is  $M = A \cdot B^{-1}$ .

### 2.3.1.d Basal Ganglia Segmentation

As the tractography of the brain is performed on the diffusion image, it inherently means that the connectivity maps and the ROI are in diffusion space. But the basal ganglia's subcortical segmentation is in T1 space. This means that even if they are registered in the same space, they will not have a voxel perfect union due to the different resolutions.

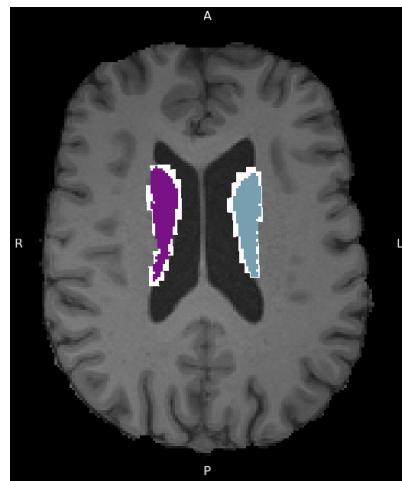


Figure 2.2: Caudate Subcortical Region Overlapping in Different Spaces

Figure 2.2 visualizes the alignment of the Caudate subcortical region, where the white (larger) region is the Basal Ganglia mask from the diffusion space and the colored (smaller) regions are the Basal Ganglia segmentation from T1 space.

In order to keep the data consistent, mapping the segmentation to the Basal Ganglia mask can be done by assigning the same label for each voxel in the basal ganglia as the label of the closest voxel in the subcortical segmentation.

### 2.3.1.e N-Dim Array

The used NIfTI format stores the raw voxel space and the affine transformation matrix separately, in order to not loose data in the process of interpolating voxels when applying the transformation. But in order to consistently compare voxel data across different spaces (even if they are registered in the same space), the transformation needs to be applied, computing the interpolated voxels in the common space, bringing them into the same raw format of matching X, Y and Z dimensions, and discarding the stored affine matrices.

By default, the native anatomical space's origin is near the center of mass of the brain, between the ears. This makes sense for medical professionals, when working with MRI records, but data structure-wise an array is indexed from 0, meaning that, after applying the transformation to the voxel space, the yielded array will only contain one quadrant of the record, as the rest are clipped in the negative regions. Thus, the space also needs to be translated with the negative vector of the transformed space's bounding box's lower end.

The translation value can be calculated by calculating the boundaries of the transformed space's bounding box. Get all 8 corners of the voxel space and apply the transformation matrix to all of them. Then get the min-max coordinates along X, Y and Z from the 8 transformed vectors, yielding the lower and upper bounds of the transformed space's bounding box.

It is very important to use the same translation value across different spaces to properly align them in the native space. For example let  $D$  and  $T$  denote a diffusion and t1 records and  $M_D$  and  $M_T$  denote their respective transformation matrices. Let  $T_D$  and  $T_T$  denote their respective translation values. In order to properly align them we need to apply  $A_D = (M_D \cdot \textcolor{red}{T}_D)$  matrix and  $A_T = (M_T \cdot \textcolor{red}{T}_D)$  matrix to  $D$  and  $T$  respectively, with matching  $\textcolor{red}{T}_D$  translation values.

The last issue is the missaligned shape of the dimensions of the T1 and diffusion records. This can be simply fixed by truncating the excess along each dimension.

### 2.3.1.f Uniform Shape

After aligning the data into the same space per record, it is still very likely that the individual records will not have a uniform shape. This is because, given that they are in native space, some records will contain a smaller volume brain, some will contain a larger, and they will not be the same.

Due to the per-voxel based prediction model architecture, this is not a problem, but fixing so that we can use the data in a spatial model like a FCNN can be simply solved by adding padding to the records in order to match their shapes.

## 2.3.2 Quality Control

Having a low count of records means that if there are even just a few outliers, it can heavily affect the end result. Thus all data were manually inspected to make sure they are as clean as possible.

### 2.3.2.a Mismatched Data

Looking through the diffusion, diffusion\_fa and diffusion\_md images, 2 records' FA and MD images were seemingly from completely different patients. Thus the FA and MD images were omitted for 2 records.

### 2.3.2.b Garbled Data

Looking through the subcortical segmentation of the Basal Ganglia revealed that 1 record had a garbled segmentation. Thus, said basal\_seg image was omitted for 1 record.

And one record had a garbled T1 FNIRT warp field. Said record was entirely omitted from the normalized set of records.

### 2.3.2.c Missing Data

Looking through the relative connectivity and streamline images, 3 records were missing these images, said 3 records were completely omitted as these records are effectively missing the labels.

Furthermore, the t1t2 images were missing for 10 records, but these were not omitted completely as the t1 images were present for these records, thus experiments only concerning the t1 can have a bit more available data.

## 2.3.3 Radiomics Features

Extracting the voxel-based radiomic features requires the tuning of two main parameters, namely the bin width and the kernel width. The binning parameter(s) influence how the intensity values of the image are binned, while the kernel size influences the size of the 'sliding window' similar to a convolution.

The two approaches for binning are absolute discretization and relative discretization. In the former, a fixed bin width is chosen and in the latter, a fixed number of bins are chosen and the bin width scales in relation to the min-max voxel values. The absolute discretization consistently provides statistically significantly more reproducible features than the relative discretization. [31] Relying on this information, the obvious choice to start with is the absolute discretization.

The bin width and the kernel width will be tuned in later experiments, and features calculated with different settings will be concatenated and used simultaneously to improve the results. The default values used will be 25 and 5 for the bin and kernel widths, respectively.

The types of radiomic features described in Table 2.2 will be used:

Feature Type	Number of Features
first order	18
gray level co-occurrence matrix (GLCM)	23
gray level size zone matrix (GLSZM)	16
gray level run length matrix (GLRLM)	16
neighbouring gray tone difference matrix (NGTDM)	5
gray level dependence matrix (GLDM)	14
3D shape	17

Table 2.2: Radiomic Feature Types

### 2.3.3.a Voxel and non-Voxel-Based

The 92 features shown in Table E.1 are voxel based calculated. Voxel-based calculation does not make sense for shape features, as it would just describe the shape of the used kernel, which is constant and independent from the input image. However, the additional shape features in Table

E.2 do make sense for the non-voxel based features, as they can be computed for each target region, both hemispheres of the ROI and the entire brain.

### 2.3.4 Coordinates

One additional input that can be included in the experiments is the coordinates. Although this approach only makes sense in normalized space, where the images from different records are aligned. This theoretically would allow the model to learn certain anatomical markers based on the location of the voxel, adding a type of global context to the input of the model.

Furthermore, this approach can be adopted to the native space, by constructing the normalized coordinate map and then 'de-normalizing' them with an inverse FNIRT warp field.

### 2.3.5 Data Augmentation

The only data augmentation that makes sense involves applying small rotation values to the input images in their native space before calculating radiomic features. Applying transformations to the already extracted features is illogical, as interpolating between voxels in feature space is unlikely to yield the same results as computing features after transforming the input images. In summary, any spatial data transformations should be performed upstream. Furthermore, data augmentation only makes sense in native space, as by definition such transformations would make the normalized image pointless.

### 2.3.6 Scaling and Normalization

As the extracted features have very different ranges, it makes sense to follow the standard practice of scaling the data to a fixed range. Inspecting the histogram of some of the radiomic features reveals that most of them follow a bell curve with moderate standard deviation, as in the example shown in Figure 2.3 (Firstorder Energy).

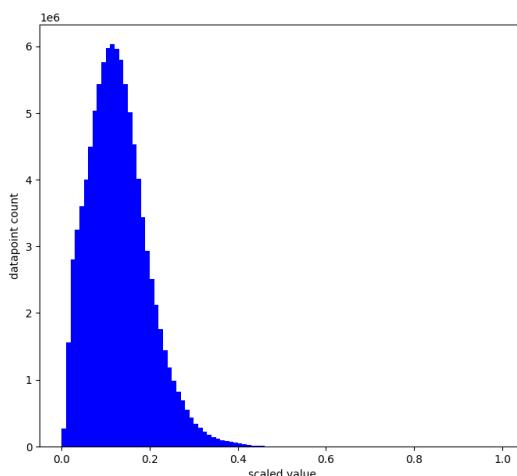


Figure 2.3: Histogram: Firstorder Energy

However, some other features such as those illustrated in Figure 2.4 (GLDM Small Dependence High Gray Level Emphasis) and Figure 2.5 (NGTDM Busyness) have a very skewed distribution,

the latter one being the most extreme case. This skewing can be mitigated by applying logarithm to the offending features.

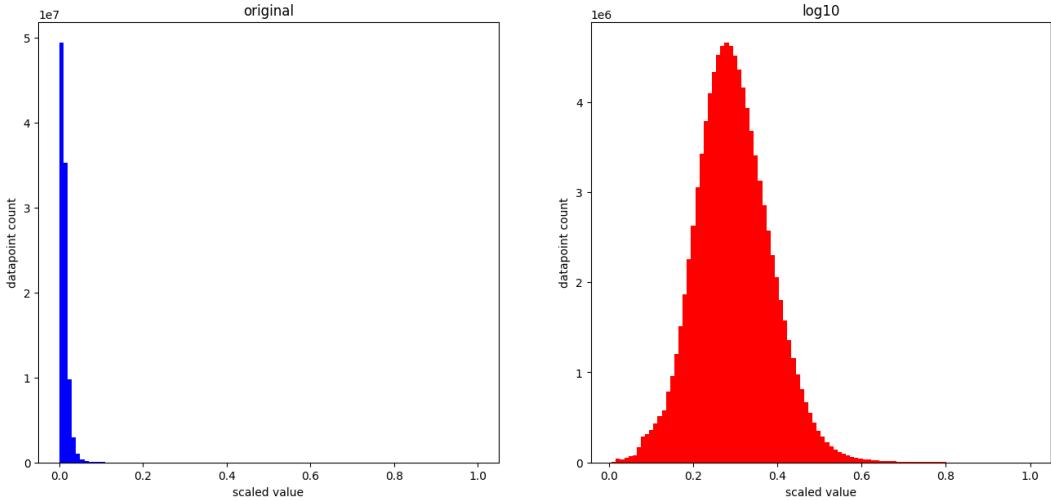


Figure 2.4: Histogram: GLDM Small Dependence High Gray Level Emphasis

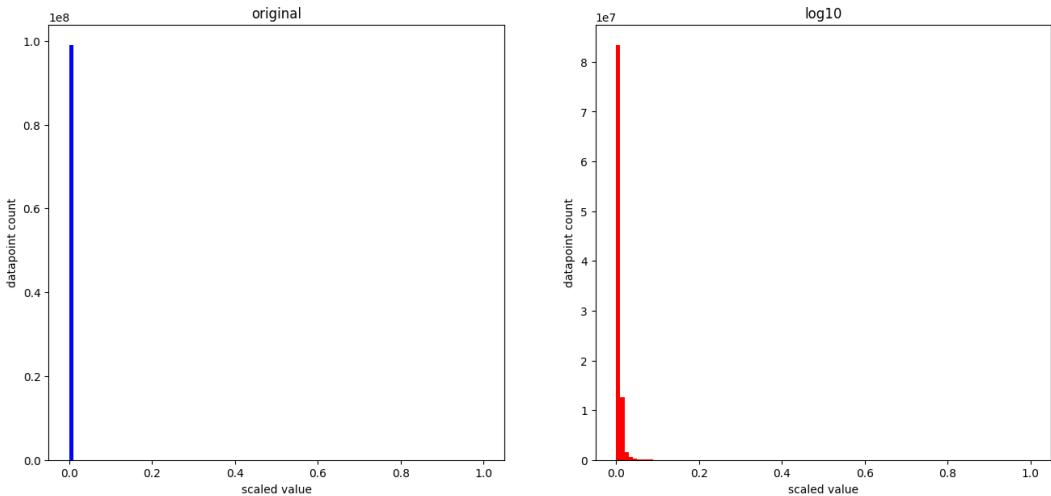


Figure 2.5: Histogram: NGTDM Busyness

Besides the standard benefits of making the optimization process more stable and efficient, and reducing the sensitivity to outliers, this also have some less evident benefits.

Although it is very subtle, but storing these records in float16 inherently loses some information. This loss is not a problem for the features that have a healthy distribution, but in the more extreme cases it can cause compression artifacts visible even to the naked eye, such as the very subtle loss of detail in Figure D.1. And in the most extreme case it can even render the entire feature useless like in Figure D.2. While the normalized features have no problem storing this fine detail in float16.

This makes the system much more robust from a practical perspective as, depending on the hardware, some GPUs are much more efficient at computing in float16; it also halves the memory and storage requirements as, in float32, a single MRI image of 92 volumes (for the 92 features) takes up around 1GB of space.

Selecting which features need normalization is done programmatically, and the exact selection criteria is detailed in Appendix C.4.

### 2.3.7 Data Balancing

Working with highly unbalanced data can be challenging, and fully balancing them does not necessarily help the model's generalization capability. Thus, a method for partially balancing the data will be used, where the bins of the unbalanced data will be up-sampled by a ratio of the difference of the number of datapoints in the bin (compared to the bin with the maximum number of datapoints). Figures 2.6 and 2.7 show how a ratio 1 means perfectly balanced data, while a ratio 0 means unbalanced data, and how the ratios in between approximately preserve the shape of the distribution and partially balance the data.

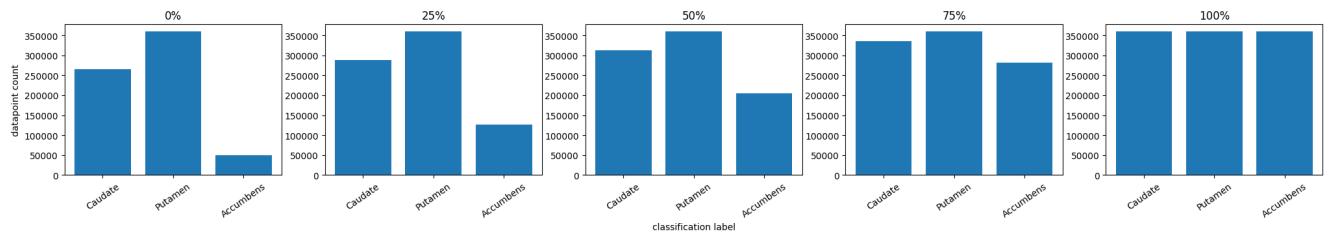


Figure 2.6: Partial Data Balancing: Subcortical

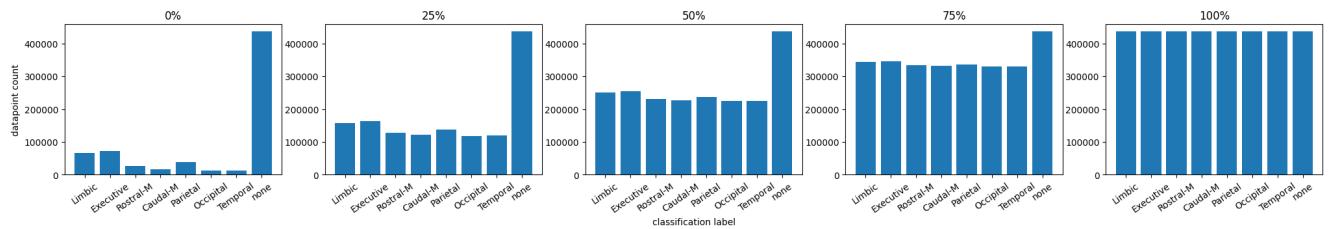


Figure 2.7: Partial Data Balancing: Relative Connectivity

For the diffusion\_fa and diffusion\_md, which are regression problems and have continuous labels, binning can be used to create artificial groups which can be balanced, as illustrated by Figures 2.8 and 2.9.

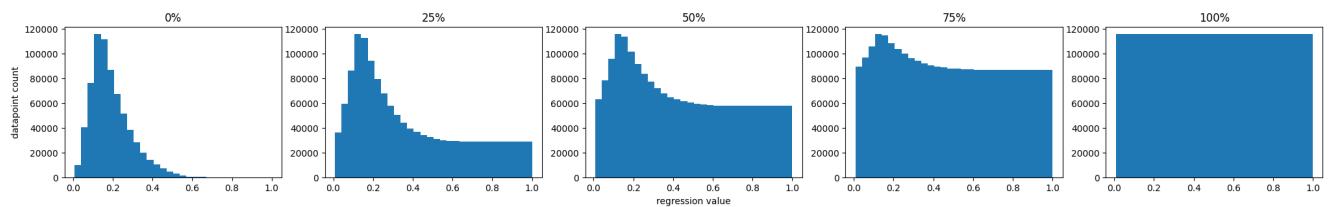


Figure 2.8: Partial Data Balancing: Diffusion FA

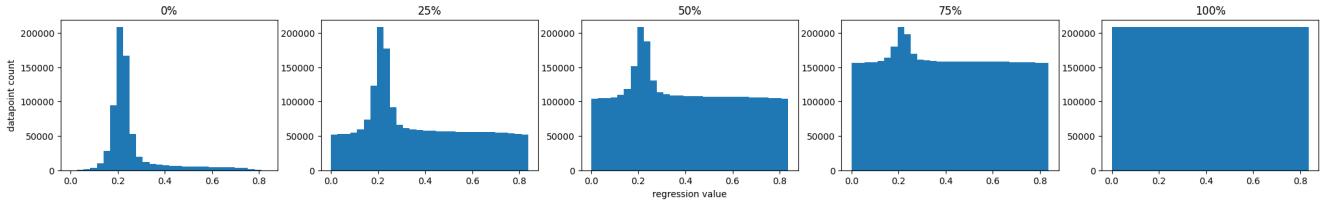


Figure 2.9: Partial Data Balancing: Diffusion MD

### 2.3.8 Clinical Data

There are additional clinical data available for the Patient records. Disease severity can be characterized in terms of CAG Age Product (CAP) score, providing a measure of cumulative exposure to the mutant HTT gene [32]. This widely accepted and used CAP score is available for all patients.

Another, newer metric for characterizing disease severity is the composite Unified Huntington's Disease Rating Scale (cUHDRS) [33], which is calculated from 4 other basic metrics: Total Functional Capacity, Total Motor Score, Symbol Digit Modalities Test and Stroop Word Reading. These are available for most patients, with a handful of exceptions.

Also, there are a total of 91 available clinical features, with a relatively high level of missing data in some of them. Moreover, there are 8 additional patients available in the clinical data. These can be used to aid the data imputation for the missing values, and can be omitted afterwards, as these have no corresponding MRI records.

All clinical features were scaled the range of 0-1 with min-max scaling (per feature) and the Euclidean distance was used for the following imputation process. The imputation strategy itself consisted of two steps: first, the few missing cUHDRS values were imputed from the CAP score and, then, the remaining features were imputed from the combined CAP and cUHDRS values.

### 2.3.9 Relative Connectivity

Relative Connectivity describes the ratio of the number of streamlines going into each cortical target. This means that the Streamline record can be converted into Relative Connectivity by simply dividing by the total number of streamlines in each voxel. With the additional inbetween step of filtering some noise, by thresholding the streamlines at 250 (5% of the total 5000 streamlines), meaning any voxel which has less than 250 streamlines to a target region are set to zero.

Then the relative connectivity could be converted into a label, by picking the label of the cortical target to the highest connection per voxel. However this would yield very noisy labels, as these ratios can be quite balanced between the multiple cortical targets, for example voxels with ratios of 0.31/0.29/0.3/0.1. To mitigate this, the relative connectivity can be thresholded at a value higher than 0.5, meaning a label can only be picked for a voxel if at least half of the connections are going to a single target.

But this also means that there can be voxels without labels. This can be dealt by introducing an artificial 'Not Connected' label for these voxels.

To achieve the best results and filtering, 0.6 was chosen for the thresholding value, as it also filters potential 50-50 situations and only allows labeling strong connections.

## 2.4 Evaluation

### 2.4.1 Train, Validation and Test Splits

There are two important aspects to consider when splitting the data into Train/Validation/Test groups. In order to truly validate the model's generalization capability, the split must happen at a record level and not at a datapoint level. This means that our model can only learn from certain records, and it can be validated on records that it has never seen before, not even partially. As a consequence, the split will not follow the defined ratio at a datapoint level, as it could happen that, by pure chance, the train split contained records with larger volumes, resulting in having more datapoints than the validation split.

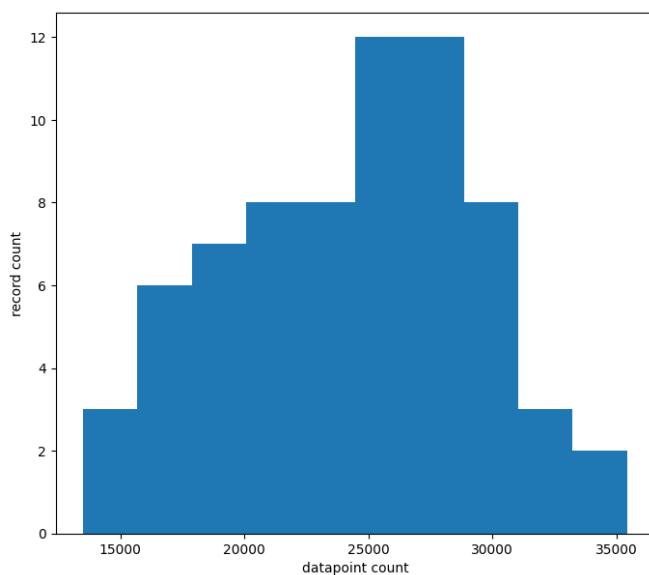


Figure 2.10: Distribution of Records in Relation to Datapoints

In practice the lower end of datapoint count per record is around half of the higher end. Figure 2.10 shows the distribution of both Control and Patient records and both Left and Right datapoints. During experimentation, with 0.8 Train split and 0.5 Validation/Test split, the datapoint ratios stayed in the range of  $0.8 \pm 0.02$  and  $0.5 \pm 0.06$  for the two split ratios.

Furthermore, and to avoid introducing bias in the case of experiments with mixed Control and Patient records, the ratio of Controls/Patients must be kept constant across the different splits. This is required as Controls and Patients can have vast differences due to neurodegeneration. An extra caveat is having another ratio that must also be kept constant for the same reason, which is the symptomatic and asymptomatic patients, as they can also have vast differences due to different stages of neurodegeneration.

### 2.4.2 Accuracy and Pearson Correlation

There are three groups of metrics for evaluating each model. First is the 'raw' (will also be referenced as 'train') metric group, which is computed on the datapoints that were extracted with the same hyperparameters as the datapoints during the training process. This means that this metric group best reflects how the model performs on the different splits, such as if the model was

trained with a balancing of 0.5, all splits will be balanced the same way and the metrics will be computed on a datapoint level (the same way as how it is naturally computed in the loss function).

The second and third groups are for comparing model performances between models and are for practical evaluation. The difference is that the metrics in this case are computed for each record, and then averaged out between records. This means that they are computed at a record level instead of at a datapoint level, resulting in the elimination of potential bias coming from the deviation from the number of datapoints per records. It also means that these metric groups will inherently ignore data balancing, as it operates on a record level.

The second group of metrics is computed in native space, while the third is computed in normalized space. This means that if the model operates in native space, the normalized metrics will be computed by predicting the datapoints for each record, then the spatial record is reconstructed from the datapoints and warped to normalized space, and then the datapoints are extracted from the normalized spatial prediction and compared against the normalized labels (this process would be computationally quite expensive, so the implementation does not follow this exact logic, but it does the same numerically; more information on this in Appendix C.3). This way, the models can have comparable metrics even if they operate in different spaces.

# 3 Experiments

The hyperparameters listed in Table 3.1 were kept constant during all of the experiments:

Hyperparameter	Value
Train Split	0.8
Validation/Test Split	0.5
Model Type	FNN
Optimizer	Adam

Table 3.1: Constant Hyperparameters

## 3.1 Subcortical Segmentation

Before addressing the primary objectives of predicting FA, MD, and Relative Connectivity, a precursor experiment was conducted, focusing on segmenting the Basal Ganglia into its subcortical regions: Caudate, Putamen, and Accumbens. This experiment served as a sanity check to validate the feasibility of the overall project. It was intentionally crafted to be simple.

Given the simplicity of the problem, extensive tuning was unnecessary, as it worked very well almost from the start. The set of hyperparameters reported in Table 3.2 remained constant during these experiments:

Hyperparameter	Value
Control/Huntington Records	Control Only
Left/Right Hemisphere Datapoints	Both
Space	Native
Image	T1
Scaling/Normalization	Normalized Voxel Based Features
Hidden Layers	1024 → 512 → 256 → 128
Loss	Categorical Crossentropy
Activation	Sigmoid (softmax for the output layer)
Learning Rate	0.001
Batch Size	10000
Early Stopping Patience	7

Table 3.2: Initial Hyperparameters

The reasoning behind the initial choices of these parameters is straightforward. The T1 image and native space were chosen, because those are the simplest to acquire in practice. Thus, if the model performs well with those, there is no need for more complicated inputs. Including both hemispheres would hopefully result in a model which can generalize better. Only using control datapoints should translate into less variance between the general characteristics of the datapoints, as it does not contain patients with neurodegeneration. The number and sizes of the hidden layers

were chosen based on the potential size of the input layer, which should range from 92 (single set of voxel based features) up to  $\sim 1000 - 2000$  including many different kernel sizes and non-voxel based features as well. The choice of the categorical crossentropy loss function and the output layer's softmax activation function are standard practices for a classification problem. The Sigmoid activation function should work fine without having to deal with exploding gradients and dieing relu problems. The default learning rate of the Adam optimizer in TensorFlow is chosen, while a batch size of 10,000 seems appropriate for a train split size of 1,000,000 datapoints. The early stopping patience of 7 epochs should also be good enough to prevent overfitting and stop the training in time, but it will be evaluated on the basis of the learning curves and the accuracy of the model.

The metric used for evaluating model performance on the Train/Validation/Test splits is Accuracy. The 'k' and 'b' notations stand for kernel and bin, where k5 means a kernel width of 5mm, b25 means an absolute bin size of 25, and b10r means relative binning with 10 bins. In the case of multiple kernel sizes denoted by a dash, it naturally only means odd kernel sizes. Table 3.3 reports the results of the subcortical experiments:

	<b>Experiment</b>	<b>Train</b>	<b>Val</b>	<b>Test</b>	<b>Input Layer</b>
1.	<b>Voxel Features k5_b25</b>	68.9	69.1	72.4	92
2.	<i>Voxel Features k5_b25</i> <b>Non-Voxel Features of Target Regions b25</b>	73.2	68.9	72.5	1576
3.	<i>Voxel Features k5_b25</i> <b>Non-Voxel Features of ROI b25</b>	75	74.3	78.5	304
4.	<i>Voxel Features k5_b25</i> <i>Non-Voxel Features of ROI b25</i> <b>Non-Voxel Features of Brain b25</b>	74.5	70.4	70.3	410
5.	<i>Voxel Features k5_b25</i> <b>Non-Voxel Features of ROI b10 b25 b50 b75</b>	71.2	70.6	74.3	856
6.	<b>Voxel Features k5_b25 - k21_b25</b> <i>Non-Voxel Features of ROI b25</i>	94.5	94.1	95.1	1040
7.	<b>Voxel Features k5_b25 - k21_b25</b>	95	94.6	93.7	828
8.	<i>Voxel Features k5_b25 - k21_b25</i> <i>Non-Voxel Features of ROI b25</i> <b>Balance Ratio 0.5</b>	94.9	94.4	94.9	1040
9.	<i>Voxel Features k5_b25 - k21_b25</i> <i>Non-Voxel Features of ROI b25</i> <b>Balance Ratio 1</b>	95.9	95.5	95.9	1040

Table 3.3: Hyperparameter Tuning: Subcortical Experiments

The best performing model was that of experiment 9, which achieved a near 96% accuracy with practically no overfitting. The biggest improvement during the experiments was to include many different kernel sizes for the voxel based features. The additional non-voxel based features of the ROI yielded only a small improvement, while balancing the data yielded a marginal improvement, by reducing overfitting.

Examples of the true/predicted records can be found in Figures D.4 - D.6, with their color coding in Table D.1. And the loss training curves can be found in Figure D.3.

## 3.2 Methodology

The experimentation from this point on, will be divided into 4 main groups:

- Native - T1
- Native - T1/T2
- Normalized - T1
- Normalized - T1/T2

These groupings enable a comprehensive investigation aligned with the state-of-the-art approaches discussed in Section 1.2. By examining these four experimental groups, this study aims to systematically evaluate the feasibility of synthesizing DTI related images. This approach provides a thorough basis for understanding the impact of imaging modalities and spatial normalization on the proposed methodology's performance.

### 3.2.1 Native or Normalized Space

The use of native and normalized spaces is essential for addressing certain variances inherent in the dataset. Working in normalized space helps reduce participant specific differences and minimizes the effects of neurodegeneration related changes, such as volumetric variations in the basal ganglia.

Normalization also mitigates minor misalignments between DTI and anatomical MRI records that might occur in native space, despite optimal affine registration. Such discrepancies can arise due to independent data acquisition processes, physiological movements, or independent preprocessing steps.

Nevertheless, the impact of these factors on model performance is not straightforward. The complex interplay between these sources of variance and their influence on the outcome necessitates experimentation to determine whether native or normalized space offers a performance advantage or whether the choice has no significant effect.

### 3.2.2 T1 or T1/T2 Imaging

The inclusion of T1 imaging is justified by its widespread historical availability and integration into many clinical protocols. On the other hand, the exploration of T1/T2 imaging stems from recent studies highlighting its correlation with myelin content. These studies suggest that the T1/T2 ratio may serve as a 'bridge' between anatomical MRI and DTI data, potentially enabling a more reliable extraction of structural connectivity information.

### 3.2.3 Further Hyperparameters

The same set of core experiments will be run for all 4 groups, and some additional experiments will be run per group, depending on how they perform. The experiments will consider the following aspects:

- Single/Many Different Kernel Sizes for Voxel Based Features

- Additional Non-Voxel Based Features
  - Single/Many Different Bin Sizes
- Control/Patient/Both Records
- Left/Right/Both Hemisphere Datapoints
- Additional Clinical Features for Patient Records
- Additional Coordinate Map Features
- Scaled Voxel Based Features (not normalized)
- Different Bin Sizes for Voxel Based Features
- Different Balance Ratios
- Data Augmentation in Native Space

These aspects facilitate a comprehensive exploration of standard data science principles, enabling the optimization of structural connectivity image synthesis. Additionally, the approach incorporates robustness assessments by experimenting with different configurations, such as splitting or combining Control/Patient records, and Left/Right hemisphere data points. Moreover, it incorporates experimental factors such as additional clinical data for patients, which may help address the increased variance within these records.

### 3.2.4 Missing Records

In order to be completely fair when comparing model performances, only records which are available for all 4 groups of experiments should be used. In practice, the records reported in Table 3.4 were missing:

Record	Missing Amount
Normalized	1
T1/T2	10
Diffusion FA & MD	2

Table 3.4: Missing Records

This meant that for the Diffusion FA & MD experiments there were a total of 13 records omitted, yielding 57 records in total, out of which 29 are Control and 28 are Patient records. For the Relative Connectivity experiment, 11 records were omitted, yielding 59 records in total, out of which 30 are Control and 29 are Patient records.

As additional experiments for the groups with more available data (such as T1, where 10 more records could be included), these records can be appended to the train split on the best performing model, which could increase the model's generalization capability and performance.

### 3.2.5 Architecture Tuning

For the best performing model, the architecture will be further tuned, considering the following aspects:

- Number of Layers and Layer Sizes
- Activation Function
- Batch Size
- Learning Rate
- Dropout Normalization
- Early Stopping Patience

## 3.3 Diffusion Fractional Anisotropy Regression

All the results of the FA experiments can be found in Tables E.3 - E.7. The baseline starting experiment tried to predict the FA from a single set of voxel based radiomic features, with a kernel size of 5 and with the same starting hyperparameters (Table 3.2) that were also used in the subcortical segmentation (with exception of the used loss function, which is Mean Squared Error instead of the Categorical Crossentropy).

The next few experiments were trying to determine how does each set (target regions, ROI, and entire brain) of non-voxel based features affect the model performance. The observations were more or less consistent between the 4 different groups of experiments (Native-Normalized & T1-T1/T2), with the final consensus being that the inclusion of the entire brain's non-voxel based features are yielding the best results, with an improvement of 0.05-0.10 in correlation compared to the baseline. Including many different bin sized non-voxel based features worsened the model performance by 0-0.03.

The biggest improvement occurred with the inclusion of many different kernel sized voxel-based features, with an improvement of 0.10-0.15. And, surprisingly, after removing the non-voxel based features, T1 experiments performance further improved by 0.01-0.02, while worsening the T1/T2 experiments by 0-0.01.

The experiments consistently showed the model performing much better on the Control records (by 0.05-0.10), as compared to the Patient records, with much less overfitting and better correlation.

The inclusion of the clinical features behaved inconsistently between the 4 groups of experiments. For the native T1, the inclusion of the CAP and cUHDRS features marginally improved the model performance, and for the normalized T1/T2 it improved model performance by 0.04-0.05, while for the native T1/T2 and normalized T1, it worsened the model performance by 0.05-0.10. The overall Patient records even with the best performing clinical features, were still performing worse than the Control records.

As expected, mixing Control and Patient records did perform worse than Control records only, but only with 0.01-0.05 correlation.

The inclusion of coordinates, did not affect the T1 models' performance, but it did marginally increase the T1/T2 models' performance.

The use of only min-max scaling, and not normalizing the datapoints, resulted in marginally worse performance.

Increasing the bin size for the voxel based radiomic features marginally decreased the model performance.

Balancing the data was a bit inconsistent between the groups of experiments, but the balance ratio of 1 usually resulted in a marginally worse, and a balance ratio of 0.5 resulted in a marginally better performance.

Adding the 10 extra T1 records to the training split for the T1 experiment only resulted in a marginal improvement for the native space, and a 0.02 improvement for the normalized space.

After combining all of the best configurations, the best performing model was the T1 normalized model, with Control records only, and with added T1 records, without any additional non-voxel based features. It reached a final correlation of **0.844/0.846/0.828** for the train/val/test splits in native space, and **0.846/0.849/0.829** in normalized space.

Tuning the model architecture by searching different layer sizes and numbers, activation functions, dropout normalization, adjusting learning rate and batch size, only increased the model's overfitting, without any actual benefits.

Examples of the true/predicted records can be found in Figures 3.2 - 3.4. And the loss training curves can be found in Figure 3.1.

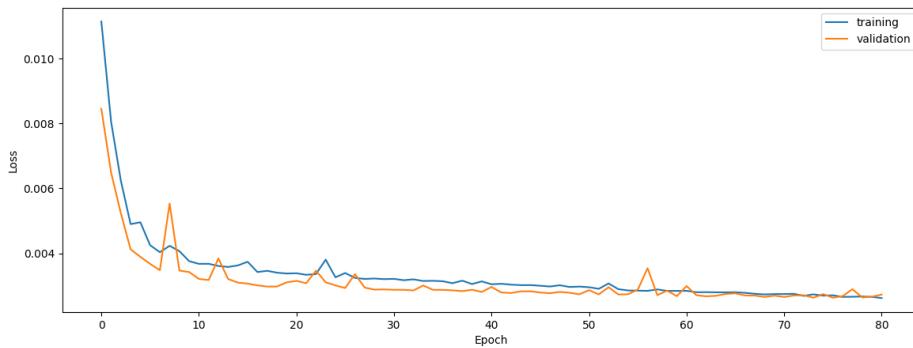


Figure 3.1: Training Curve: Diffusion Fractional Anisotropy Best Performing Model

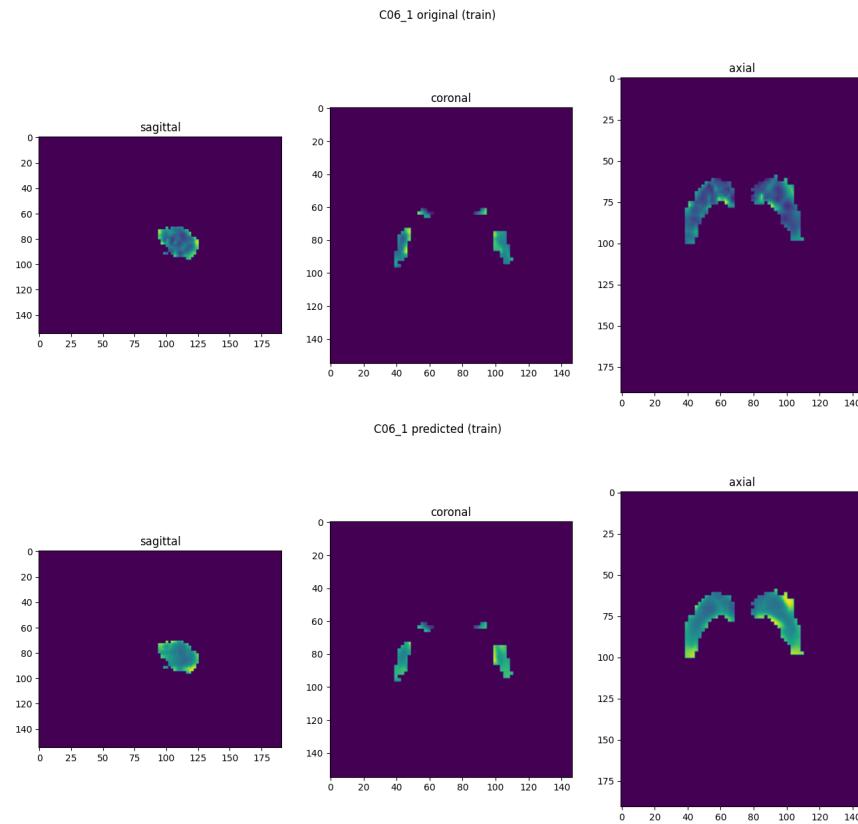


Figure 3.2: Train Predictions: Diffusion Fractional Anisotropy Best Performing Model

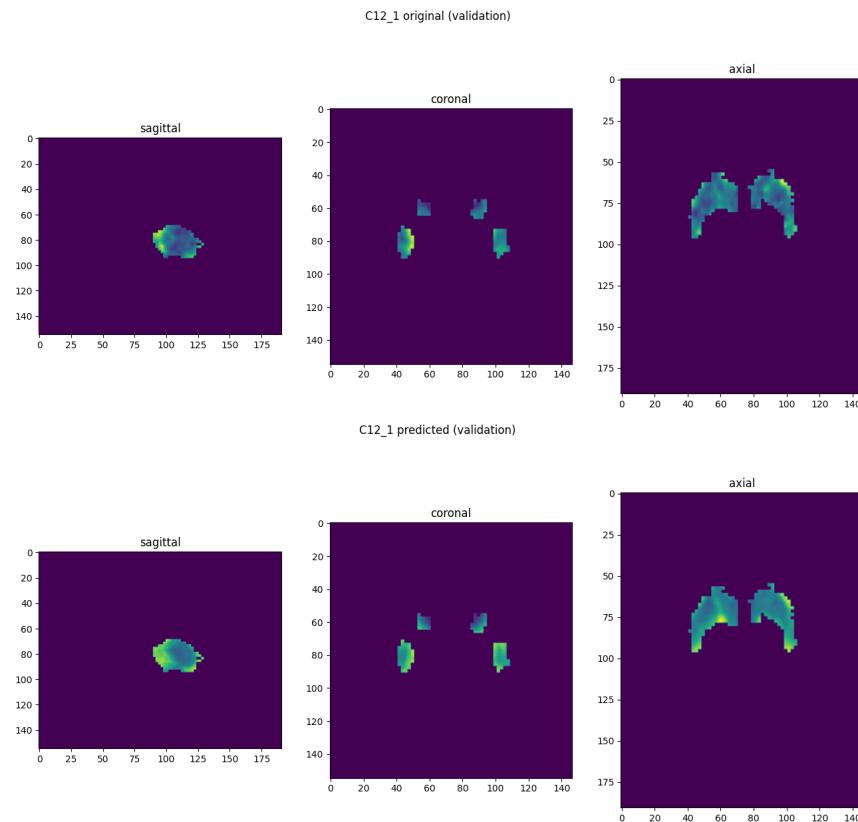


Figure 3.3: Validation Predictions: Diffusion Fractional Anisotropy Best Performing Model

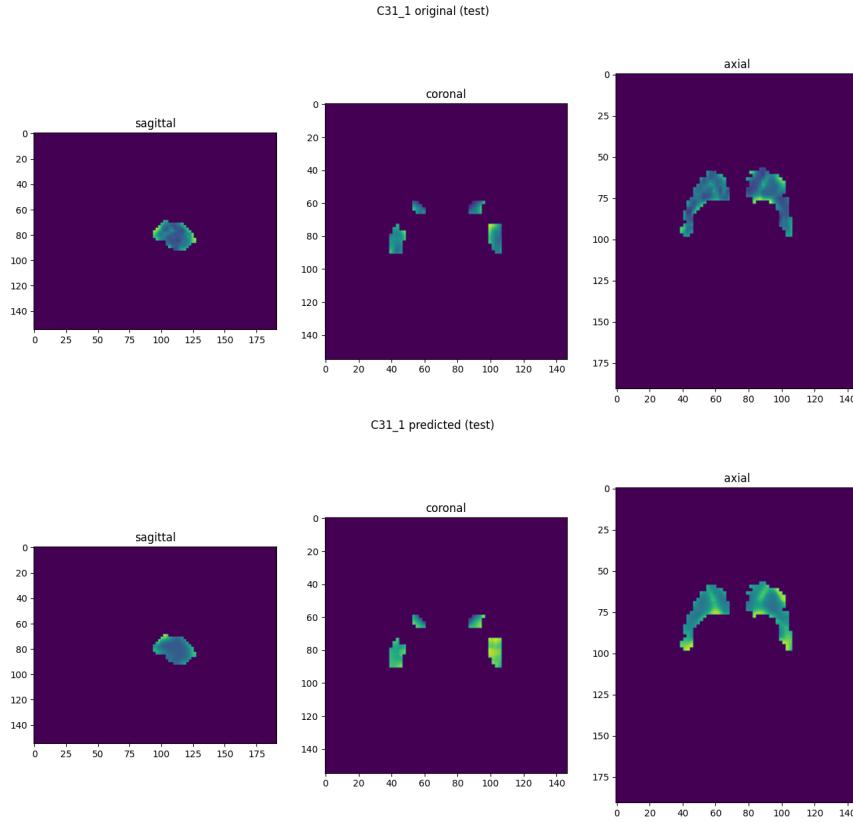


Figure 3.4: Test Predictions: Diffusion Fractional Anisotropy Best Performing Model

### 3.4 Mean Diffusivity Regression

A similar set of experiments were run for predicting MD. The compilation of results is available from Tables E.8 - E.12. These experiments were showing strong performance from the onset: even the baseline Native T1 experiment with a single set of voxel based features resulted in a correlation of 0.94 without any overfitting.

No significant observation can be made here, besides the Patient records performing marginally worse.

The best performing model was Native T1, on the Control records only, with the additional non-voxel based features of the entire brain, and many different voxel based kernel sizes. It reached a final correlation of **0.947/0.955/0.951** for the train/val/test splits in native space, and **0.954/0.957/0.963** in normalized space.

Tuning the model architecture, by searching different layer sizes and numbers, activation functions, dropout normalization, adjusting learning rate and batch size, did not increase the performance, not even for the train split, indicating that this is the absolute best this model can do.

Examples of the true/predicted records can be found in Figures 3.6 - 3.8. And the loss training curves can be found in Figure 3.5.

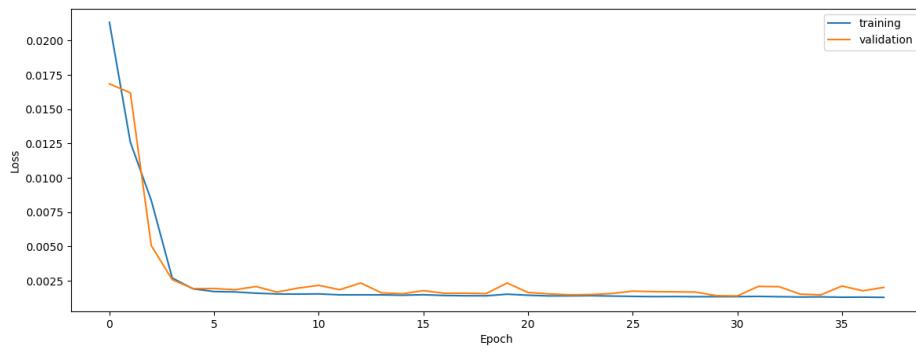


Figure 3.5: Training Curve: Mean Diffusivity Best Performing Model

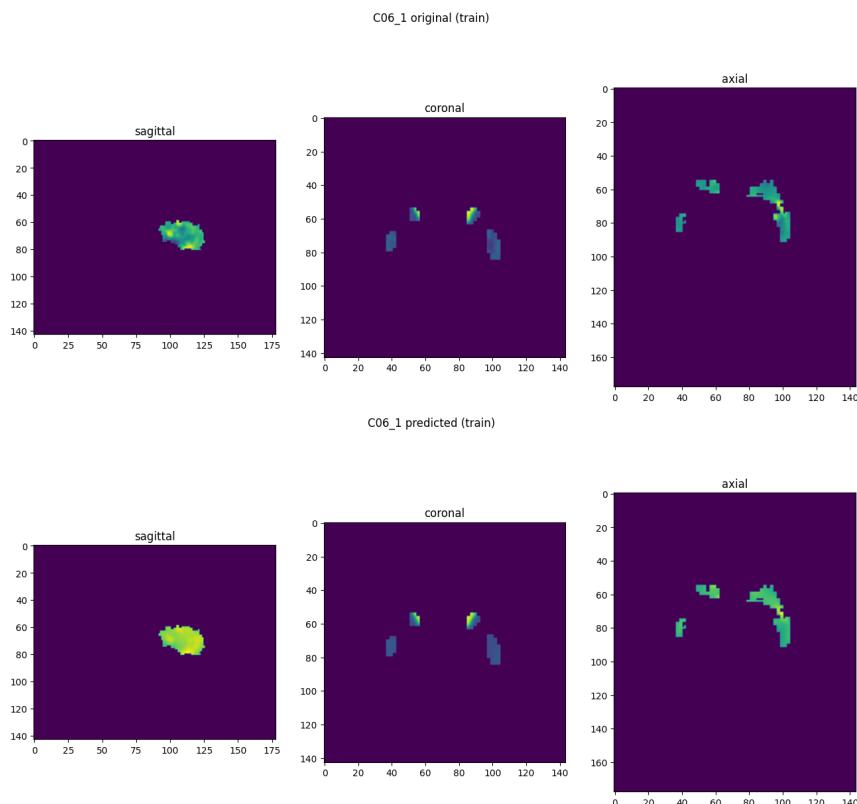


Figure 3.6: Train Predictions: Mean Diffusivity Best Performing Model

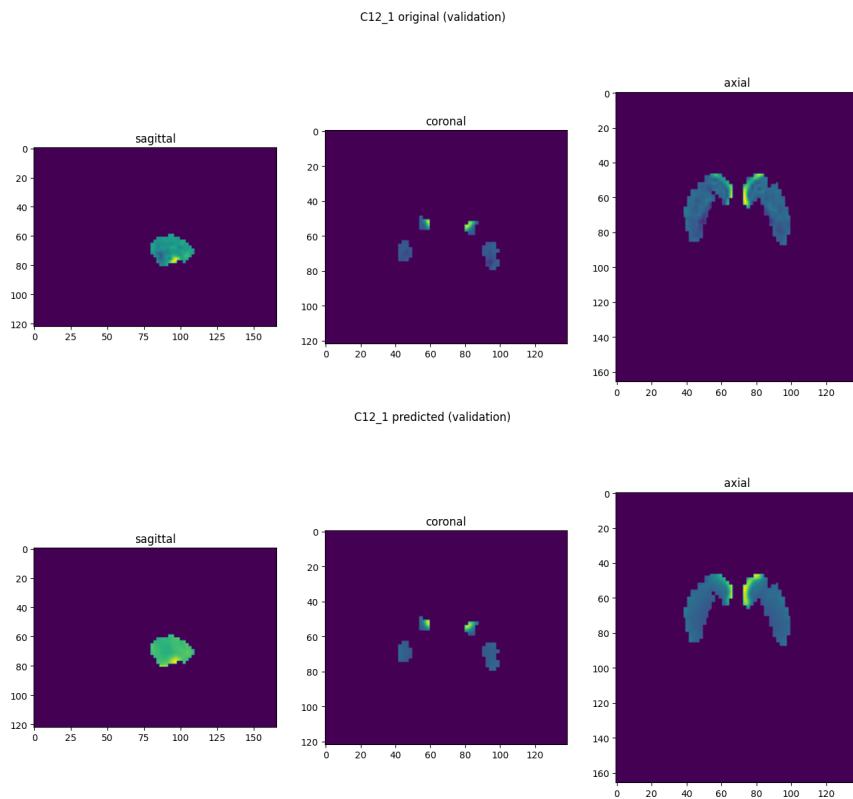


Figure 3.7: Validation Predictions: Mean Diffusivity Best Performing Model

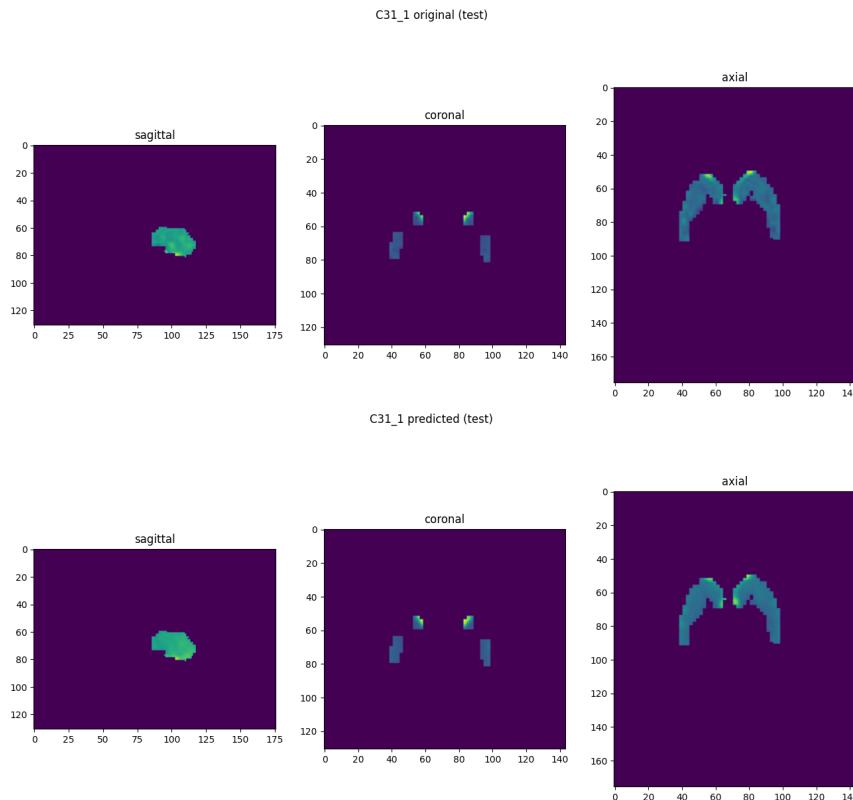


Figure 3.8: Test Predictions: Mean Diffusivity Best Performing Model

### 3.5 Relative Connectivity Segmentation

All numerical results corresponding to the Relative Connectivity Segmentation experiments can be found in Tables E.13 - E.17. The baseline starting experiment is trying to predict the Relative Connectivity (preprocessed with the method described in subsection 2.3.9) from a single set of voxel based radiomic features, with a kernel size of 5, and with the same starting hyperparameters (Table 3.2) that were also used in the subcortical segmentation.

The subsequent experiments aimed to determine how does each set (target regions, ROI, and entire brain) of non-voxel based features affect the model performance. The results were fairly consistent between the 4 different group of experiments (Native-Normalized & T1-T1/T2), with the final consensus being that the inclusion of non-voxel based features does not improve model performance.

The biggest improvement (5-10%) was obtained with the inclusion of many different kernel sized voxel-based features.

The experiments consistently showed the model performing much better on the Control records, compared to the Patient records, with much less overfitting and better accuracy (by 2-5%).

The inclusion of the clinical features yielded inconsistent results between the four groups of experiments. For the normalized experiments, it mostly had no effect, or made it marginally worse, while for the native T1 experiments, the inclusion of CAP and cUHDRS features marginally improved the model performance. Finally, for the native T1/T2, it increased train/validation performance substantially, but yielded worse test performance, due to strong overfitting.

Mixing Control and Patient records only performed marginally worse than using only Control records.

Including coordinates, consistently increased accuracy by 1-2%.

Only using min-max scaling, and not normalizing the datapoints, resulted in marginally worse performance.

Increasing the bin size for the voxel based radiomic features marginally decreased the model performance.

Balancing the data consistently and substantially decreased performance (depending on the balance ratio by 5-20%)

The addition of the 10 extra T1 records as part of the training split for the T1 experiments did not affect the model performance.

After combining all of the best configurations, the best performing model was the T1 normalized model, with Control records only, with the additional coordinate inputs, without any additional non-voxel based features. It reached a final accuracy of 72.6/72.1/73.3 for the train/val/test splits in native space, and 73.2/71.9/73 in normalized space.

After tuning the model architecture, by searching different layer sizes and numbers, activation functions, dropout normalization, adjusting learning rate and batch size, the only thing which marginally increased model performance was lowering the batch size to  $10^3$  and lowering the learning rate to  $10^{-4}$ , yielding a final accuracy of **73.3/72.9/73.4** in native space, and **73.5/72.3/73.4** in normalized space.

As these numbers can be misleading due to the highly unbalanced data, the best way to get more insight on how the model is performing is by observing the confusion matrices in Figure 3.9 and Figure 3.10. Matrices in Figure 3.9 are normalized along the predicted label axis, effectively displaying the precision in the diagonals; in Figure 3.10, they are normalized along the true label

axis, effectively displaying the recall in the diagonals. The first and most evident observation is that the unbalanced nature of the data is reflected on the confusion matrix, as the over-represented 'not connected' datapoints have a much better precision and recall than the rest.

Also, the model is more effective at minimizing false positives than it is at minimizing false negatives, since it generally has a higher precision than recall for practically all labels (except the 'not connected').

Examples of the true/predicted records can be found in Figures 3.13 - 3.15. And the loss training curves can be found in Figure 3.11. The cortical targets of the brain can be seen in Figure 3.12, with their corresponding color coding in Table 3.5, with matching color coding in the example true/predicted records.

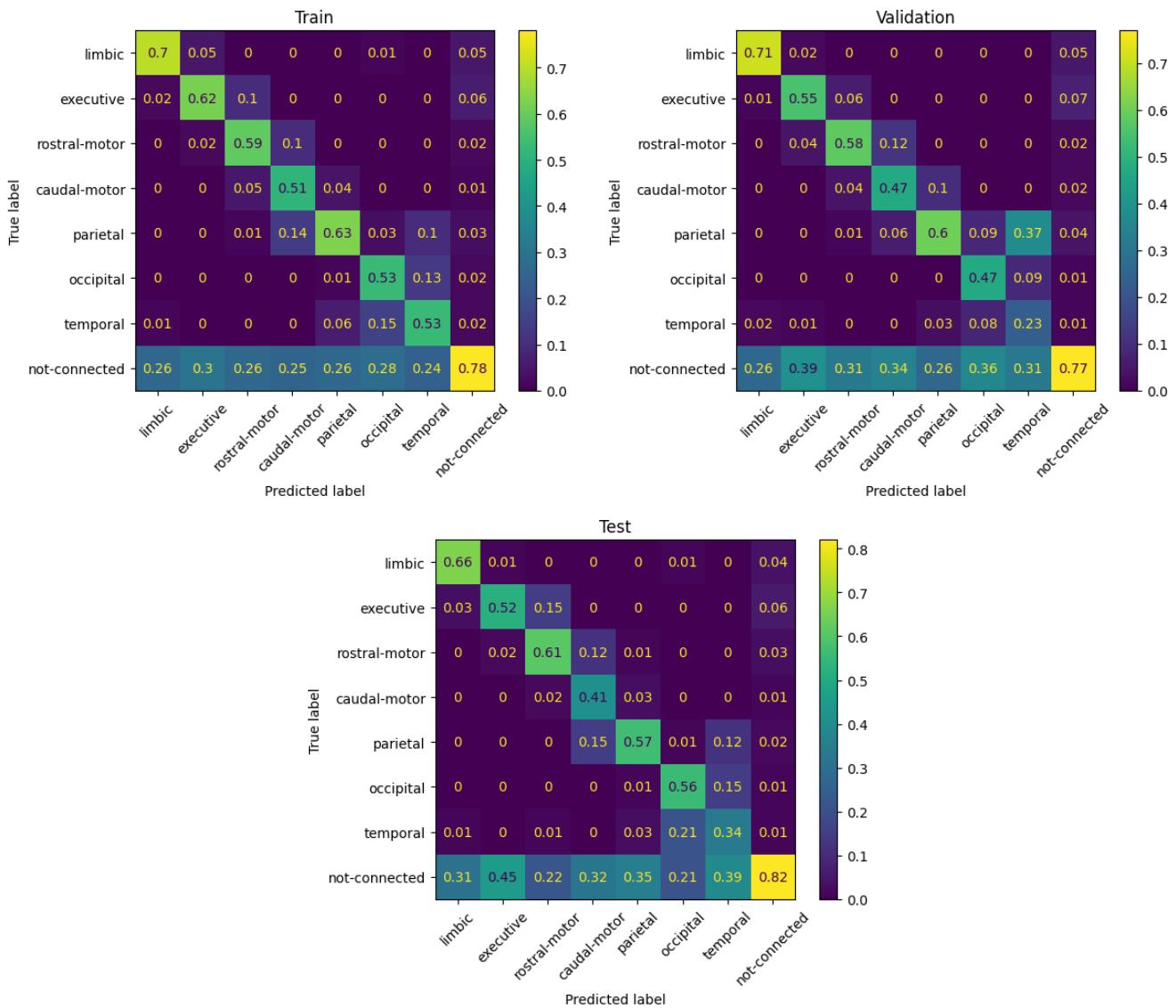


Figure 3.9: Confusion Matrices (Precision): Relative Connectivity Best Performing Model

These matrices also tell that not all labels are performing equally. As the model clearly struggles with the recall of 'temporal' target region datapoints.

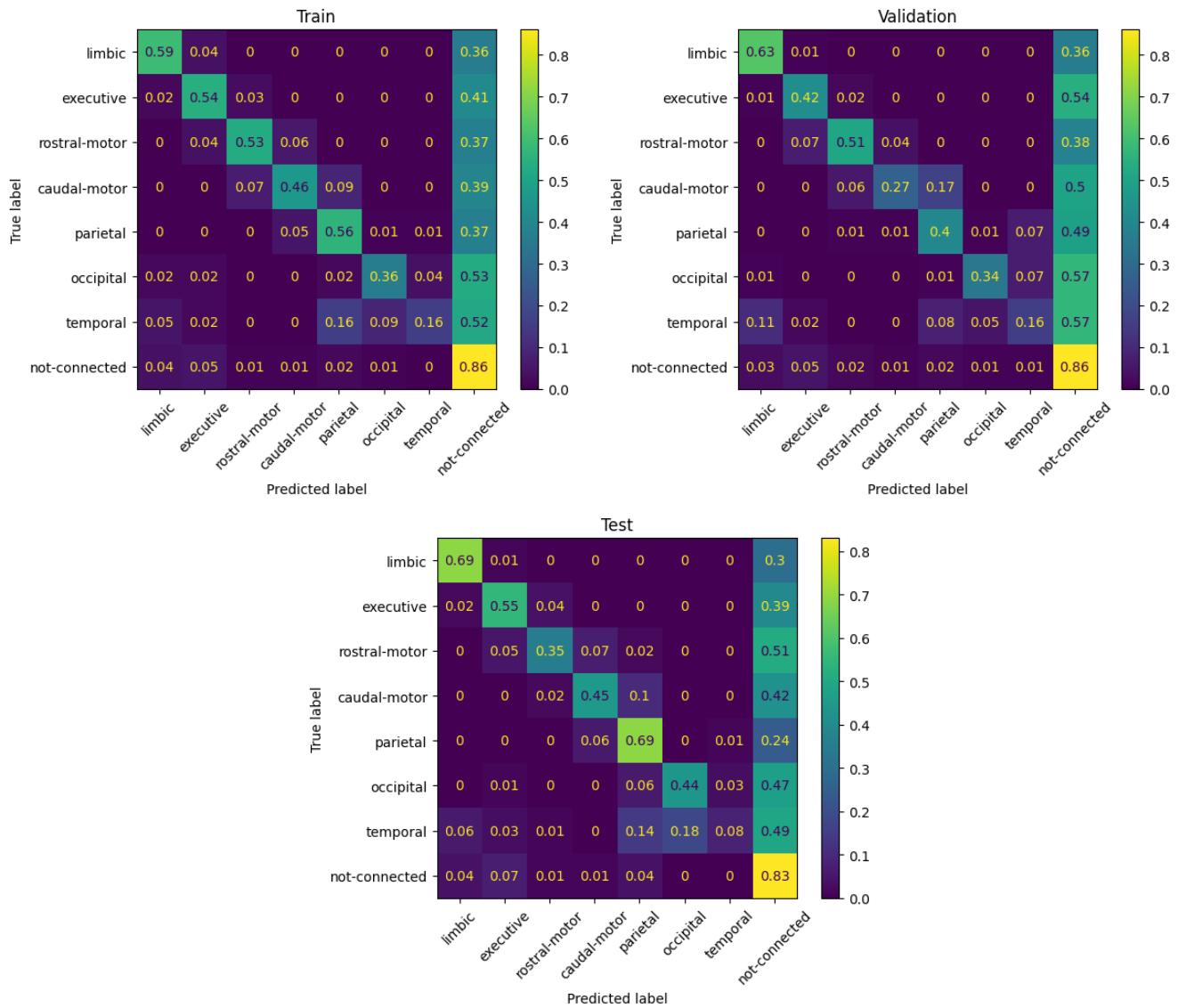


Figure 3.10: Confusion Matrices (Recall): Relative Connectivity Best Performing Model

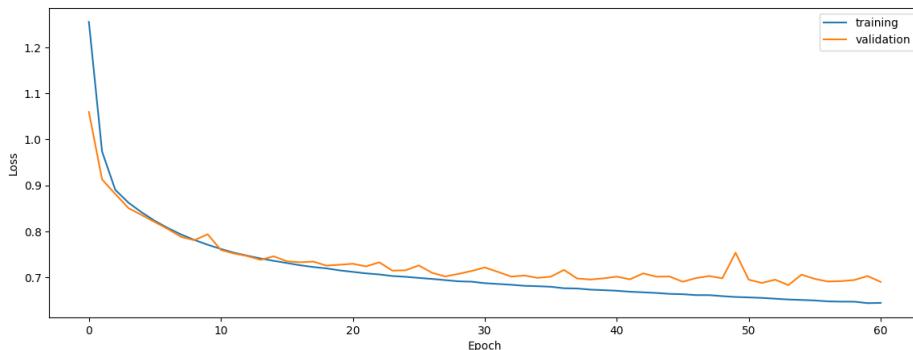


Figure 3.11: Training Curve: Relative Connectivity Best Performing Model

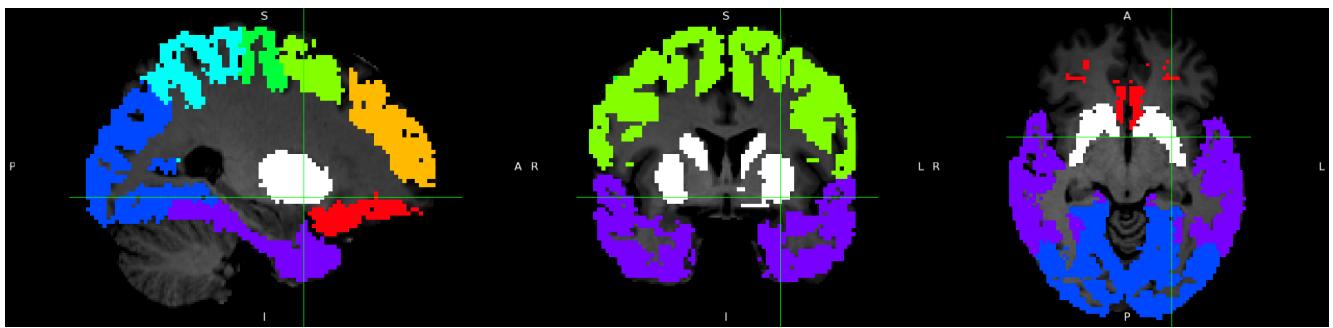


Figure 3.12: Basal Ganglia (ROI) &amp; Cortical Targets

Color	Region
Red	Limbic
Orange	Executive
Light Green	Rostral-Motor
Green	Caudal-Motor
Light Blue	Parietal
Blue	Occipital
Purple	Temporal
Pink	Not Connected

Table 3.5: Cortical Targets Color Coding

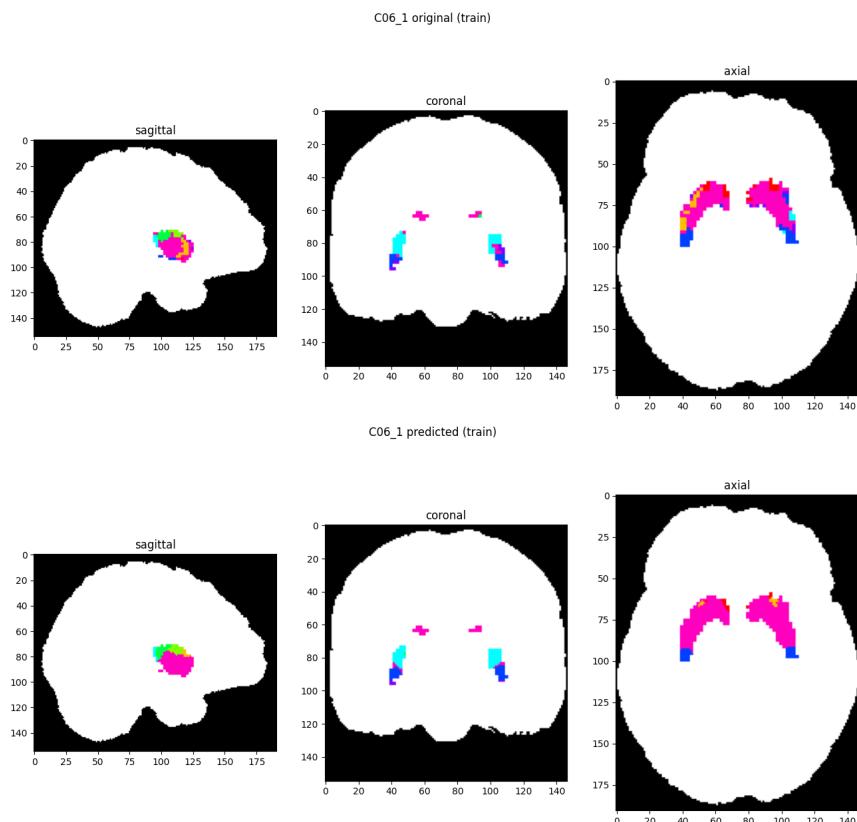


Figure 3.13: Train Predictions: Relative Connectivity Best Performing Model

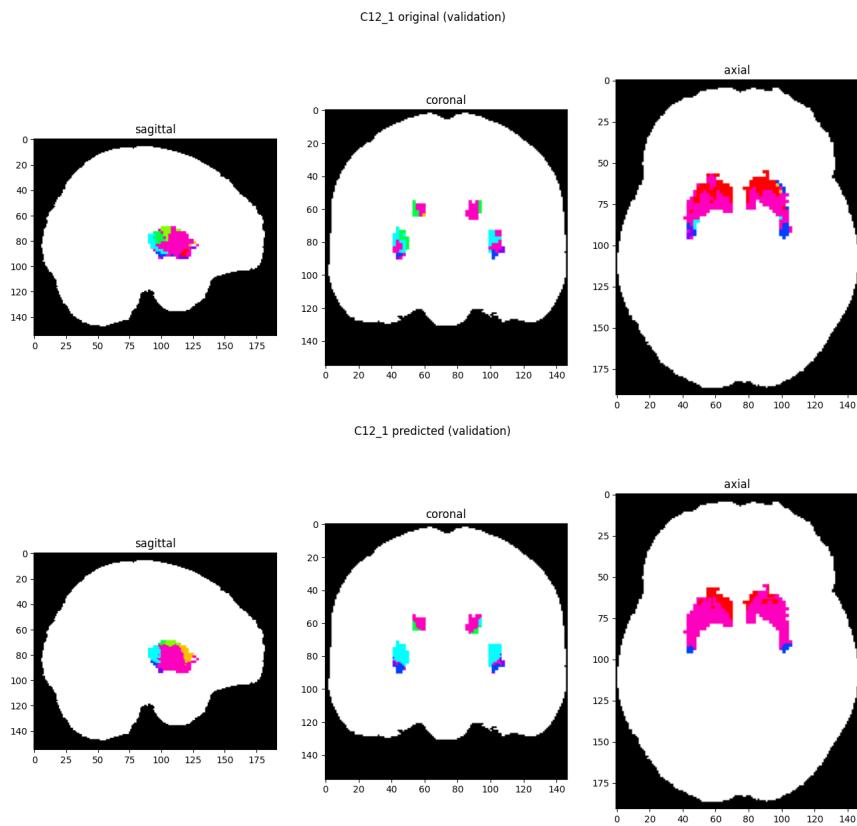


Figure 3.14: Validation Predictions: Relative Connectivity Best Performing Model

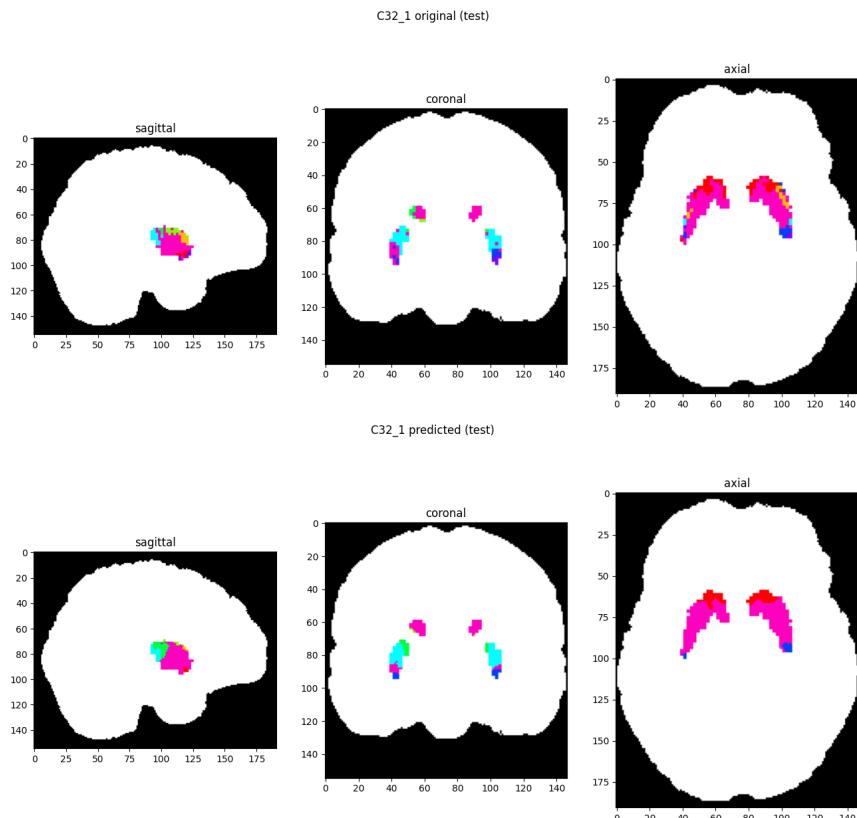


Figure 3.15: Test Predictions: Relative Connectivity Best Performing Model

### 3.5.1 Exhaustive Sequential Backwards Feature Selection

Exhaustive sequential backwards feature selection concerns training the model iteratively by removing a single feature at a time, going through all features. After an iteration is completed, the best performing model is chosen, and the corresponding feature is permanently removed before the next iteration, where the model goes through the remaining  $n - 1$  features.

The stopping criterion can be varied from setup to setup, but, in this case, the evaluation metric was chosen to be the raw accuracy, using as a stopping point that where all runs in the iteration performed worse than the baseline (the model with all features included) by more than 2%.

Feature selection was only ran for the Relative Connectivity Segmentation problem, due to it being time consuming and computationally intensive, given the time constraints of the thesis development. This was executed on a network cluster, where the cluster head would give out tasks (a task being a model to train with a set of features to exclude), and the workers would return the validation accuracy.

Two logistical oversights that were not mitigated in time were incurred. The first mistake being that the execution of the feature selection was rushed due to the longevity of the task, and the rest of the hyperparameters (besides the excluded features) were chosen before finishing the basic experimentation. And the second being an indirect consequence to it being rushed: the usage of a defective code for balancing the data, where the bug was found only after already executing the feature selection.

Due to these mistakes, the selected features turned out not to be very efficient with the final hyperparameters. The feature selection was not executed again due to time and resource limitations. More information detailing this mistake and oversight can be found in the Future Improvements 5.1 subsection.

Nevertheless, the expected result of re-running the feature selection would be something similar to the current sub-optimal result, where around one third of the features could be excluded before stopping and a maximum increase of 2% obtained in validation accuracy. The first features to be excluded (See Table 3.6) were predominantly from the first order radiomics feature class, followed by the GLCM radiomics feature class.

Iteration	Excluded Feature	Accuracy
0.	BASELINE	59.0
1.	firstorder_MeanAbsoluteDeviation	59.6
2.	firstorder_Entropy	59.9
3.	firstorder_Energy	58.7
4.	glszm_SmallAreaLowGrayLevelEmphasis	58.5
5.	firstorder_90Percentile	59.3
6.	glcm_Autocorrelation	59.5
7.	firstorder_Mean	59.4
8.	firstorder_Maximum	59.1
9.	glszm_LowGrayLevelZoneEmphasis	58.9
10.	glcm_Imc1	60.0
11.	firstorder_Median	60.1
12.	glszm_ZoneVariance	59.3
13.	firstorder_TotalEnergy	58.9
14.	firstorder_10Percentile	60.4
15.	firstorder_Minimum	59.1
16.	glszm_SmallAreaHighGrayLevelEmphasis	59.9
17.	firstorder_InterquartileRange	60.0
18.	firstorder_Kurtosis	59.6

19.	firstorder_RobustMeanAbsoluteDeviation	59.3
20.	firstorder_RootMeanSquared	59.8
21.	gldm_LargeDependenceEmphasis	58.9
22.	firstorder_Variance	59.1
23.	glcm_DifferenceAverage	59.4
24.	firstorder_Uniformity	58.9
25.	firstorder_Skewness	59.0
26.	firstorder_Range	58.6
27.	glcm_JointAverage	58.7
28.	glcm_ClusterProminence	59.2
29.	glcm_ClusterTendency	58.5
30.	glcm_ClusterShade	60.2
31.	glcm_Correlation	60.0
32.	glcm_DifferenceEntropy	58.1
33.	glrlm_RunVariance	59.3
34.	glcm_JointEnergy	59.6
35.	glcm_Contrast	60.5
36.	glcm_Idm	59.3
37.	glcm_Imc2	58.7
38.	glcm_DifferenceVariance	59.2
39.	glcm_Idmn	58.0
40.	glcm_MCC	57.9
41.	glcm_JointEntropy	56.6

Table 3.6: Feature Selection: Excluded Features

### 3.5.2 Streamline Regression

An alternative analytical approach would be to try predicting the raw streamline images, and then processing the predictions in the same way the relative connectivity labels were computed in the first place. This method could result in a more robust solution for predicting the upstream data, before preprocessing.

Training seven expert models for the seven streamline images (for each cortical target), with the same hyperparameters as the best performing model from the relative connectivity segmentation yielded underwhelming results, with the computed accuracies being 65.1/65.3/63.8, and very low precision/recall for all labels (except the 'not connected'), which is explained by the predicted labels mainly being 'not connected', in any case. This approach was not explored further in depth due to limited time and resources.

# 4 Sustainability

## 4.1 Environmental Aspect

### 4.1.1 Developement

The biggest environmental impact of this project was the required computational power to train the models. Over the entire duration of the project, approximately 5,000 models were trained.

As part of it, the most demanding phase was the exhaustive sequential backward feature selection, which in itself required the training of 3,000 models. The return value of this computation is close to none, as it only marginally increased model performance. The only real benefit is the model being able to reach the same performance using two thirds of the original features. This is a marginal result, because the omitted features belong to different radiomics feature groups, and the biggest overhead for most feature groups is calculating the voxel-based matrix volume, meaning that if not all features are omitted from a feature group, it only decreases the computational requirements marginally.

The 3,000 models for feature selection were trained on a wide variety of GPUs in a network cluster. These GPUs were H100, P40, RTX3060, RTX3090, RTX2080, multiple T4s (Google Collab's), multiple P100s (Kaggle's). Mining the cluster head's log, reveals that around half of these models were trained on the H100 and the other half on the rest. The H100 was consuming around 300 Watts of power while training with a speed of 5s/epoch and taking 30-60 epochs on average to stop. The H100 consumed between  $1500 \cdot 0.3\text{kW} \cdot 30 \cdot 5\text{s} \div 60 \div 60 = 18.75\text{kWh}$  on the lower and  $2 \cdot 18.75 = 37.5\text{kWh}$  on the upper end. The other half is harder to estimate due to the many different GPUs used in the cluster, but a crude estimate is  $1500 \cdot 0.2\text{kW} \cdot 30 \cdot 8\text{s} \div 60 \div 60 = 20\text{kWh}$  on the lower and  $2 \cdot 20 = 40\text{kWh}$  on the upper end. Rounding the sum of them up to 80kWh and adding an extra 30% overhead for running the rest of the components of the servers/PCs, results in an upper boundary of 104kWh consumed during the feature selection.

The remaining 2,000 models were trained on the P40 and RTX3060, with the P40 training approximately three fourths of them. Doing the same estimations for the P40 yields  $1500 \cdot 0.2\text{kW} \cdot 30 \cdot 7\text{s} \div 60 \div 60 = 17.5\text{kWh}$  on the lower and  $2 \cdot 17.5 = 35\text{kWh}$  on the upper end. For the RTX3060, it yields  $500 \cdot 0.15\text{kW} \cdot 30 \cdot 7\text{s} \div 60 \div 60 = 4.375\text{kWh}$  on the lower and  $2 \cdot 4.375 = 8.75\text{kWh}$  on the upper end. Rounding their sum up to 45kWh and adding an extra 30% overhead for running the rest of the components of the server, results in an upper boundary of 58.5kWh consumed during the the rest of the trainings.

Feature extraction run on a CPU, with the entire server consuming 200 Watts during the process. The entire duration for the feature extraction run was around 2 weeks. Its total power consumption was  $0.2\text{kW} \cdot 14\text{d} \cdot 24\text{h} = 67.2\text{kWh}$ .

In summary, this project had the environmental impact summarized in Table 4.1, for an average of 0.25€/kWh for the price of the electricity [34] and 0.2kgCO<sub>2</sub>e/kWh for the emission [35]:

Part	Consumption	Price	Emission
feature selection	104kWh	26€	20.8kgCO <sub>2</sub> e
rest of the experiments	58.5kWh	14.6€	11.7kgCO <sub>2</sub> e
feature extraction	67.2kWh	16.8€	13.4kgCO <sub>2</sub> e
<b>Total</b> (+10% overhead; rounded up)	260kWh	65€	52kgCO <sub>2</sub> e

Table 4.1: Sustainability: Power Consumptions and CO<sub>2</sub> Emissions

For reference, this number is a bit less than the average Spanish household's monthly power consumption of 324kWh. [36]

### 4.1.2 Production

This project, whether implemented in a production, clinical or research setting, has the potential to significantly enhance sustainability in its respective fields. Acquiring DTI data typically requires up to 40 minutes, depending on the specific parameters, whereas obtaining a T1 image takes only about 8 minutes, making data acquisition a considerable time-saving step. Additionally, DTI data demands extensive and labor-intensive preprocessing. **Accelerating clinical or research workflows by a factor of 5** not only saves time but also increases the capacity for data collection or patient processing by the same factor.

Synthesizing a single record of relative connectivity or diffusion FA/MD takes only a few seconds on a GPU, while the radiomic feature extraction takes around 20 minutes. This translates into an approximate energy demand of  $200W \cdot (5s \div 60 + 20m) \div 60 = 67Wh$  for synthesizing a record. It can be argued that the time saved with the data acquisition is lost with the time it takes to extract the radiomic features, but neither the doctor (or MRI operator) and patient need to be present during the computation of the feature extraction. Assuming that an MRI machine has a consumption of 25-70kW [37] and calculating with an average of 50kW, adds up to a power requirement of  $50kW \cdot 40m \div 60 = 33.3kWh$  per DTI record. Adding up the total energy requirements of T1 data acquisition and record synthesis is  $50kW \cdot 8m \div 60 + 67Wh \div 1000 = 6.7kWh$  per record. This means that **this method could be 5 times more sustainable than the traditional approach**, while returning the energy consumption of the development phase in the data acquisition of  $260 \div (33.3 - 6.7) = 10$  records, which is naturally a minuscule number when compared to how many MRI recordings are routinely acquired in the world.

## 4.2 Economic Aspect

### 4.2.1 Cost

The electricity cost during the development phase amounted to 65€. Approximately half of this cost was covered by colleagues and their networks, who contributed computational resources during the feature selection process. Additional support came from free services provided by Google and Kaggle.

Realistically, billing this project would consist of three components: the cost of labor, the price of the server used for development, and the electricity consumed. With the electricity cost already detailed, the server cost and labor are interconnected. The project could have been executed on a less powerful machine, though this would have required more time and more careful focus on

software design. Throughout the development, the primary limiting factors were RAM and disk storage. Significant effort was invested in developing efficient data structures and optimizing the memory footprint to address these constraints.

In conclusion, the minimum hardware which this project could be feasibly executed on is 32GBs of RAM, Intel Core i5-12600K (or similar), Nvidia RTX3060 (or similar), and 256GBs of storage. The current cost (new parts ordered from Amazon) of such server (with 'cheap' consumer grade hardware) would be around 300€ for the GPU, 150€ for the CPU, 50€ for the RAM, 25€ for the SSD, and 150€ for the motherboard, adding up to 675€. The main limiting factors in this configuration are the RAM, which was utilized up to 128GBs during development; and disk storage, which was utilized up to 300GBs. The extra RAM and storage would cost an additional 200€ and 25€ totalling at a new sum of 900€.

The development of this project required approximately 500 hours of work. Making calculations using the average hourly rate of 19€ corresponding to a software engineer in Spain [38], leads to a total labor cost of  $19\text{€} \cdot 500\text{h} = 9500\text{€}$ . Therefore, the total billing for the project would be the one summarized in Table 4.2:

Part	Price
Cumulative Salary	9500€
Server Components	900€
Electricity	65€
<b>Total</b>	10465€

Table 4.2: Project Cost Breakdown

#### 4.2.2 Return

This project is by all means a proof of concept, aiming to assess the viability of the proposed approach. More information on the exact conclusions can be found in Chapter 5, but the potential return value of this project is huge.

As stated in the previous subsection 4.1.2, this project has the potential to accelerate the clinical and/or research workflows by a factor of 5. The economic implications are applicable for both clinical and research workflows. In the clinical workflow, **this could increase the number of patients processed by a factor of 5**, with minimal extra cost (especially if compared to the alternative of buying 4 additional MRI machines and hiring operators for them). In a research workflow, it would probably have less of an economic impact, but more of a logistical one, as it could simplify the data acquisition, thus **allowing researchers to have up to 5 times more data acquired** with minimal extra resources invested. But even more importantly, it opens the door of processing past anatomical MRI records, as they are much more common than DTI records, virtually **increasing the available data for researchers by several orders of magnitude** (depending on the exact application).

## 4.3 Social Aspect

### 4.3.1 Inclusivity

Out of the potential inclusivity aspects, including age, gender, sex, and cultural diversity, it is sex the one that requires most attention, as males and females have slightly different brain structures. [39] For the control records used, there are 17 male and 15 female ones, which is a relatively even split; for the patient records, there are 25 male and 13 female records, which is relatively unbalanced. This could, in theory, negatively impact the under-represented group if there were truly any differences between the two sexes that mattered from the model's perspective. However, in this project, and due to the very limited number of records available, it was not an option to discard one third of the patients in order to have the a completely balanced dataset. Moreover, it would need further experimentation so as to determine if this aspect truly impacts the results.

This potential difference was overlooked during most of the project's lifetime, but it should have been included as another constant ratio to be kept during the train/validation/test splitting, which was covered in subsection 2.4.1. Due to limited time for thesis completion and resources, the models were not re-trained with this new rule in mind, and this is something that has been included in the Future Improvements 5.1 subsection.

The rest of the considerations are less pressing and hard to take into account, like the brain structure of different ethnicities, given that this dataset does not contain any information regarding this aspect. The only other aspect potentially under consideration should be age, but this is partially accounted for as part of the constant symptomatic/asymptomatic ratio, because being symptomatic is closely related to the CAP score (covered in subsection 2.3.8), which is directly related to age. Thus symptomatic/asymptomatic is indirectly related to age.

## 4.4 Risks

Environmental, Economic and Social risks are not really applicable to this project, as it is a foundational proof of concept research project.

# 5 Conclusions

This foundational project has huge potential thanks to the possible returns and applications of the approach proposed. Due to deployment time and resource limitations, the experiments only focused on the Basal Ganglia, so by no means any of the performance metrics prove that a generalized solution is viable. However the FA and MD predictions are really promising, and would be interesting to see how it would perform on the entire brain, and not just the ROI. On the other hand, the Relative Connectivity predictions are not even nearly precise enough to call them usable, but, admittedly, the labels themselves will inherently contain quite a bit of noise, due to the very sensitive process of extracting the labels with tractography.

This project only delved into a tiny fraction of the endless sea of possible preprocessing approaches, and it only experimented with some very basic model architectures. Nevertheless, the viability of this approach and hypothesis is not disproven, but it definitely needs the implementation of some new ideas to make it fully functional.

Key conclusions regarding different imaging modalities and spatial normalization primarily stem from the FA and Relative Connectivity experiments. The MD and Subcortical Segmentation experiments showed only marginal differences across modalities, making it difficult to draw definitive conclusions from these results.

In general, experiments conducted in normalized space slightly outperformed those in native space (by 1-2%). This could be attributed to the removal of some variance in normalized space or reduced misalignment between anatomical MRI and DTI images.

An interesting observation across many experiments was that the T1/T2 modality, despite showing marginally better training performance (by 1-2%), exhibited more significant overfitting. This resulted in similar validation performance and slightly worse test performance (by 1-2%) compared to T1. Notably, T1/T2 performed comparably or better with less input data, such as when only a single kernel size of voxel-based features were provided. This suggests that the T1/T2 modality, although not performing as well overall as T1, might harbor untapped potential, particularly, under certain conditions.

There are alternative explanations for why the results do not fully align with state-of-the-art studies that use the T1/T2 ratio as a myelin proxy. One possible factor is the non-isotropic voxel size of our T2 images, which, when combined with 3D radiomic feature extraction, may introduce noise or bias, particularly affecting the GLRLM feature class. This issue could distort feature values due to the skewed run lengths along the affected axis.

The neurodegeneration observed in patients with HD significantly impacts model performance, likely due to the higher variance in their data. This issue could potentially be mitigated by increasing the dataset size to account for this variability. However, mixing Control and Patient records resulted in only a slight performance decrease, suggesting that combining the two groups, lead to more robust models.

The inclusion of clinical features exhibited inconsistent results across experiments and modalities. While some configurations led to improved performance in Patient records, these models never outperformed those trained on Control or Control+Patient data, indicating that clinical features may not be a beneficial input in this context.

Data augmentation had minimal impact on the native space experiments, likely due to the

limited variation introduced by the mild rotations applied during augmentation. However, more aggressive augmentation could compromise applicability, as it would likely introduce unrealistic variations not typically observed in the raw MRI scans.

It was also consistently shown in the experiments, that the logarithmic data normalization on the heavily skewed features, did improve performance.

Exploring hemisphere-specific datapoints revealed an unexpected finding: the right hemisphere consistently performed better than the left. This pattern can be explained in part by the asymmetrical neurodegeneration in HD, where the left hemisphere is affected more severely. Interestingly, this trend was also present in the Control group, which warrants further investigation. Ultimately, combining both hemispheres resulted in models that generalized well, even for the left hemisphere, reaching the performance levels of models trained on right hemisphere data only.

## 5.1 Future Improvements

As mentioned in subsection 3.5.1, there was a serious oversight during the execution of the exhaustive sequential backwards feature selection. At the time it was believed that the model was performing better on balanced data, due to a bug in the code. Thus, all of these models are performing much worse than the models during the experimentation. The bug was fixed, but due to limited time and resources the feature selection was not executed again.

Further investigation of the sex imbalance issue reported in subsection 4.3.1 reveals that, with the seed used for the experiments results are not terribly imbalanced for the most part, but there is definitely room for improvement by enforcing a constant ratio. The male/(male+female) ratio for the control record splits is 0.49/0.67/0.67 (train/validation/test), while it is 0.62/0.5/1 for the patient records.

Additionally, one aspect was not investigated thoroughly over the thesis: the kernel size, binning parameters, and feature class relationships during the voxel based feature extraction. For the sake of simplicity, only a unified binning method was used for all kernel sizes and all feature classes. But, logically, the binning parameters could be optimized for each kernel size and feature class. For example, the GLRLM feature class could yield fundamentally different features even just by adjusting the bin size a tiny bit at large kernel sizes.

The selection of the most efficient kernel size combinations was not investigated thoroughly either. Due to limited resources, the maximum kernel size used was 21, but there was no reason to stop there.

Ultimately, and even with the relatively few simple aspects that were taken into consideration during this thesis, the hyperparameter space is gigantic and some compromises had to be made. After investing over 500 hours into this project, the educated guesses for these potential improvements are:

- The feature selection should have a marginal improvement on the model performance.
- The gender imbalance should not have a measurable impact on the model performance on this scale (with less than 70 available records in total).
- The binning parameter optimization for the kernel sizes and feature classes would probably have a big impact on the model performance.

And lastly, all experiments should be repeated at least a few times, with different splits and seeds. And the experiments should be evaluated with the means and medians of the model performance indicators, across different runs.

Ultimately there are two paths that could be followed in order to pursue this approach further: First, the FA and MD experiments could be expanded to the entire brain and a generalized model could be developed. The second one would entail the refinement of the Relative Connectivity experiments, as they require overall improvements and the development of new ideas.

# Sources of Information

- [1] *T1 weighted image*. URL: <https://radiopaedia.org/articles/t1-weighted-image> (visited on 01/13/2025).
- [2] *Professional Guide for Medical Professionals: T1 vs T2 MRI*. URL: [https://abiscans.com/blog/mri/professional-guide-for-medical-professionals-t1-vs-t2-mri/#Understanding\\_the\\_Technical\\_Differences\\_Between\\_T1\\_and\\_T2\\_MRI\\_Scans](https://abiscans.com/blog/mri/professional-guide-for-medical-professionals-t1-vs-t2-mri/#Understanding_the_Technical_Differences_Between_T1_and_T2_MRI_Scans) (visited on 01/13/2025).
- [3] Zhang Jiawei. “Basic Neural Units of the Brain: Neurons, Synapses and Action Potential”. In: (May 2019). URL: <https://doi.org/10.48550/arXiv.1906.01703>.
- [4] J.L. De Reuck. “White Matter”. In: (2014), pp. 759–762. URL: <https://doi.org/10.1016/B978-0-12-385157-4.01190-8>.
- [5] G. Jean Harry and Arrel D. Toews. “Myelination, Dysmyelination, and Demyelination”. In: *Handbook of Developmental Neurotoxicology* (1998), pp. 87–115. URL: <https://doi.org/10.1016/B978-012648860-9.50007-8>.
- [6] Wint Yan Aung, Soe Mar, and Tammie Ls Benzingher. “Diffusion tensor MRI as a biomarker in axonal and myelin damage”. In: *Journal of Medical Imaging* 5.5 (Oct. 2013), pp. 427–440. URL: <https://doi.org/10.2217/iim.13.49>.
- [7] Peter J. Basser and Carlo Pierpaoli. “Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI”. In: *Journal of Magnetic Resonance, Series B* 111.3 (1996), pp. 209–219. ISSN: 1064-1866. URL: <https://doi.org/10.1006/jmrb.1996.0086>.
- [8] T. E. J. Behrens et al. “Non-invasive mapping of connections between human thalamus and cortex using diffusion imaging”. In: *Nature Neuroscience* 6.7 (July 2003), pp. 750–757. ISSN: 1546-1726. URL: <https://doi.org/10.1038/nn1075>.
- [9] T.E.J. Behrens et al. “Probabilistic diffusion tractography with multiple fibre orientations: What can we gain?” In: *NeuroImage* 34.1 (2007), pp. 144–155. ISSN: 1053-8119. URL: <https://doi.org/10.1016/j.neuroimage.2006.09.018>.
- [10] Marius E Mayerhoefer et al. “Introduction to radiomics”. In: *Journal of Nuclear Medicine* 61.4 (2020), pp. 488–495. URL: <https://doi.org/10.2967/jnumed.118.222893>.
- [11] *PyRadiomics - Radiomic Features*. URL: <https://pyradiomics.readthedocs.io/en/latest/features.html> (visited on 01/10/2025).
- [12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [13] G. Bebis and M. Georgopoulos. “Feed-forward neural networks”. In: *IEEE Potentials* 13.4 (1994), pp. 27–31. URL: <https://doi.org/10.1109/45.329294>.
- [14] Christopher J.C. Burges. “A Tutorial on Support Vector Machines for Pattern Recognition”. In: *Data Mining and Knowledge Discovery* 2.2 (June 1998), pp. 121–167. ISSN: 1573-756X. URL: <https://doi.org/10.1023/A:1009715923555>.
- [15] José L Lanciego, Natasha Luquin, and José Obeso. “Functional neuroanatomy of the basal ganglia”. In: *Cold Spring Harbor perspectives in medicine* (2012). URL: <https://doi.org/10.1101/cshperspect.a009621>.

- [16] Olivia C Matz and Muhammad Spocter. “The Effect of Huntington’s Disease on the Basal Nuclei”. In: *Cureus* (2022). URL: <https://doi.org/10.7759/cureus.24473>.
- [17] Marianne J U Novak et al. “Basal ganglia-cortical structural connectivity in Huntington’s disease”. In: *Human Brain Mapping* 36.5 (Jan. 2015), pp. 1728–1740. URL: <https://doi.org/10.1002/hbm.22733>.
- [18] Zahra Shams, David G. Norris, and José P. Marques. “A comparison of in vivo MRI based cortical myelin mapping using T1w/T2w and R1 mapping at 3T”. In: *PLOS ONE* 14.7 (July 2019), pp. 1–22. URL: <https://doi.org/10.1371/journal.pone.0218089>.
- [19] Md. Nasir Uddin et al. “Comparisons between multi-component myelin water fraction, T1w/T2w ratio, and diffusion tensor imaging measures in healthy human brain structures”. In: *Scientific Reports* 9.1 (Feb. 2019), p. 2500. ISSN: 2045-2322. URL: <https://doi.org/10.1038/s41598-019-39199-x>.
- [20] Smarta Sangui et al. “3D MRI Segmentation using U-Net Architecture for the detection of Brain Tumor”. In: *Procedia Computer Science* 218 (2023), pp. 542–553. ISSN: 1877-0509. URL: <https://doi.org/10.1016/j.procs.2023.01.036>.
- [21] Xiaomeng Li et al. “H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes”. In: *IEEE Transactions on Medical Imaging* 37.12 (2018), pp. 2663–2674. URL: <https://doi.org/10.1109/TMI.2018.2845918>.
- [22] Huan Minh Luu and Sung-Hong Park. “Extending nn-UNet for Brain Tumor Segmentation”. In: (2022), pp. 173–186. URL: [https://doi.org/10.1007/978-3-031-09002-8\\_16](https://doi.org/10.1007/978-3-031-09002-8_16).
- [23] Wei Chen et al. “S3D-UNet: Separable 3D U-Net for Brain Tumor Segmentation”. In: (2019), pp. 358–368. URL: [https://doi.org/10.1007/978-3-030-11726-9\\_32](https://doi.org/10.1007/978-3-030-11726-9_32).
- [24] Mehrdad Noori, Ali Bahri, and Karim Mohammadi. “Attention-Guided Version of 2D UNet for Automatic Brain Tumor Segmentation”. In: (2019), pp. 269–275. URL: <https://doi.org/10.1109/ICCKE48569.2019.8964956>.
- [25] Zhao Zhang, Feng Feng, and Tingting Huang. “FNNS: An Effective Feedforward Neural Network Scheme with Random Weights for Processing Large-Scale Datasets”. In: *Applied Sciences* 12.23 (2022). ISSN: 2076-3417. URL: <https://doi.org/10.3390/app122312478>.
- [26] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. “Core Vector Machines: Fast SVM Training on Very Large Data Sets”. In: *Journal of Machine Learning Research* 6.13 (2005), pp. 363–392. URL: <http://jmlr.org/papers/v6/tsang05a.html>.
- [27] Samy Bengio Ronan Collobert. “SVM Torch: Support Vector Machines for Large-Scale Regression Problems”. In: *Journal of Machine Learning Research* 1 (2001), pp. 143–160. URL: <https://www.jmlr.org/papers/volume1/collobert01a.html>.
- [28] Lena Bundschuh et al. “A Novel Radiomics-Based Tumor Volume Segmentation Algorithm for Lung Tumors in FDG-PET/CT after 3D Motion Correction—A Technical Feasibility and Stability Study”. In: *Diagnostics* 12.3 (2022). ISSN: 2075-4418. URL: <https://doi.org/10.3390/diagnostics12030576>.
- [29] Constance A. Owens et al. “Lung tumor segmentation methods: Impact on the uncertainty of radiomics features for non-small cell lung cancer”. In: *PLOS ONE* 13.10 (Oct. 2018), pp. 1–22. URL: <https://doi.org/10.1371/journal.pone.0205003>.

- [30] Baderaldeen A. Altazi et al. “Reproducibility of F18-FDG PET radiomic features for different cervical tumor segmentation methods, gray-level discretization, and reconstruction algorithms”. In: *Journal of Applied Clinical Medical Physics* 18.6 (2017), pp. 32–48. URL: <https://doi.org/10.1002/acm2.12170>.
- [31] Loïc Duron et al. “Gray-level discretization impacts reproducible MRI radiomics texture features”. In: *PLOS ONE* (2019). URL: <https://doi.org/10.1371/journal.pone.0213459>.
- [32] *Special Characteristics of HD Data*. URL: <https://enroll-hd.org/for-researchers/analyzing-data/special-characteristics-of-hd-data-2> (visited on 01/15/2025).
- [33] Karl D. Kieburtz et al. “Unified Huntington’s disease rating scale: Reliability and consistency”. In: *Movement Disorders* 11.2 (1996), pp. 136–142. URL: <https://doi.org/10.1002/mds.870110204>.
- [34] *Spain - Household electricity prices*. URL: <https://countryeconomy.com/energy-and-environment/electricity-price-household/spain> (visited on 12/28/2024).
- [35] B.W. Ang and Bin Su. “Carbon emission intensity in electricity production: A global analysis”. In: *Energy Policy* 94 (2016), pp. 56–63. ISSN: 0301-4215. URL: <https://doi.org/10.1016/j.enpol.2016.03.038>.
- [36] *Electricity Consumption per Dwelling*. URL: <https://www.odyssee-mure.eu/publications/efficiency-by-sector/households/electricity-consumption-dwelling.html> (visited on 12/28/2024).
- [37] *MRI and Sustainability*. URL: <https://www.siemens-healthineers.com/perspectives/MRI-reducing-energy-consumption> (visited on 12/28/2024).
- [38] *Average Software Engineering Salaries by Country*. URL: <https://codesubmit.io/blog/software-engineer-salary-by-country> (visited on 12/28/2024).
- [39] Daphna Joel. “Male or Female? Brains are Intersex”. In: *Frontiers in Integrative Neuroscience* 5 (2011). ISSN: 1662-5145. URL: <https://doi.org/10.3389/fnint.2011.00057>.
- [40] Stephen M. Smith. “Fast robust automated brain extraction”. In: *Human Brain Mapping* 17.3 (2002), pp. 143–155. URL: <https://doi.org/10.1002/hbm.10062>.
- [41] Alexander Leemans and Derek K. Jones. “The B-matrix must be rotated when correcting for subject motion in DTI data”. In: *Magnetic Resonance in Medicine* 61.6 (2009), pp. 1336–1349. URL: <https://doi.org/10.1002/mrm.21890>.
- [42] Andri C. Tziortzi et al. “Connectivity-Based Functional Analysis of Dopamine Release in the Striatum Using Diffusion-Weighted MRI and Positron Emission Tomography”. In: *Cerebral Cortex* 24.5 (Jan. 2013), pp. 1165–1177. ISSN: 1047-3211. URL: <https://doi.org/10.1093/cercor/bhs397>.
- [43] Timothy E. J. Behrens et al. “Learning the value of information in an uncertain world”. In: *Nature Neuroscience* 10.9 (Sept. 2007), pp. 1214–1221. ISSN: 1546-1726. URL: <https://doi.org/10.1038/nn1954>.

# Appendices

## A Previous Work

### A.1 Participants

In the clinical analysis, 47 patients with HD were included, of which 24 of the gene mutation carriers were manifest (symptomatic) patients. And 23 of the gene mutation carriers were premanifest (asymptomatic) patients.

A total of 41 patients from the clinical analysis along with 32 healthy controls matched for age, sex, and years of education participated in the neuroimaging analysis. Out of these, 22 patients were symptomatic (defined as those with a diagnostic confidence level of at least 4 on the cUHDRS). The rest 19 patients were asymptomatic participants. Three symptomatic patients did not undergo tractography analysis due to missing data.

None of the participants reported previous history of neurological disorder other than HD. The study was approved by the ethics committee of Bellvitge Hospital in accordance with the Helsinki Declaration of 1975. All participants signed a written declaration of informed consent.

### A.2 Clinical Evaluation

Three clinical domains (i.e. motor, cognitive, and behavioral) were evaluated using a battery of clinical scales and questionnaires. This included the cUHDRS, which consists of motor, cognitive, and behavioral subscales [33].

### A.3 MRI Data Acquisition

MRI data were obtained using the 3T whole-body MRI scanner (Siemens Magnetom Trio; Hospital Clinic, Barcelona), with a 32-channel phased array head coil. Structural images included a conventional high resolution 3D T1 image, 208 sagittal slices, matrix  $208 \times 256 \times 256$ , repetition time 1970ms, echo time 2.34ms, inversion time 1050ms, flip angle 9°, field of view 256mm, slice thickness 1mm with no gap between slices.

Additionally, diffusion weighted MRI was obtained using a sequence with a dual spin-echo DTI with GRAPPA (reduction factor of 4) cardiac gating, with echo time 92ms, 2mm isotropic voxels with no gap, 60 axial slices, field of view 236mm. To obtain the diffusion tensors, diffusion was measured along 64 non-collinear directions, using a single b-value of  $1500 \frac{s}{mm^2}$  and interleaved with 9 non-diffusion  $b = 0$  images. Frequency selective fat saturation was used to suppress fat signal and avoid chemical shift artifacts.

## A.4 Tractography

DTI data were automatically processed using the FMRIB's diffusion toolbox (FDT) in the FMRIB software library (FSL). First, skull stripping was performed using the Brain Extraction Tool [40]. Head motion and eddy current correction were then applied, with the gradient matrix rotated accordingly [41].

A multi fibre diffusion model was fitted to the data using FDT [8], employing a Bayesian approach to estimate a probability distribution function for the principal fibre direction at each voxel, accommodating crossing fibres within each voxel [9].

In addition, affine intra subject non-linear transformation to an MNI152 template were calculated using FNIRT, for both the diffusion weighted and the anatomical images. The diffusion tensor was reconstructed using a standard least squares tensor estimation algorithm for each voxel, and FA and MD, as an index of microstructural organisation map was calculated from corresponding eigenvalues and eigenvectors.

## A.5 Connectivity Calculation

Cortical target regions were provided by Tziortzi [42] and consisted of Limbic, Executive, Rostral-Motor, Caudal-Motor, Parietal, Occipital and Temporal regions. All 7 regions were transformed from the normalized MNI152 T1 space to the participants' native diffusion space via the FNIRT warp fields.

The basal ganglia were segmented in the native T1 space into lateralised caudate, nucleus accumbens, and putamen using the FIRST toolbox in FSL.

Following this, the segmented basal ganglia subcortical regions were warped to the participants' native diffusion space following the same process as the cortical targets. The nucleus accumbens, caudate, and putamen were then combined to create ROIs for each participant, to be used for connectivity based parcellation.

Connectivity based hard parcellation of the basal ganglia was carried out using the "find the biggest" algorithm in FSL [43], using the cortical classifiers. Specifically, the probtrackx2 function from the FDT toolbox was employed using standard settings (number of streamlines 5000, number of steps per sample 2000, step length 0.5mm, and curvature threshold 0.2). All images were in the native diffusion space, with the seed as the basal ganglia and classification targets set as 14 cortical targets (7 targets x 2 hemispheres, left and right) [8]. This resulted in the streamline image of the basal ganglia being segmented into 14 lateralised maps with regards their connectivity to each of the 7 cortical targets.

These connectivity based probabilistic streamline images had a set of values for each voxel of the basal ganglia which represents how many tractography samples out of the default setting of 5000 reach the cortical targets. These images were then thresholded at 5% to remove noise and aberrant connections. Following this, relative connectivity maps were then calculated by dividing individual probabilistic images by the sum of all images, and then thresholding at 50% to minimize any overlap between these maps. These relative connectivity maps allowed for the measurement and comparison of changes in of topographically organized connectivity.

# B Software Design

The software design was mainly guided by the provided format of the raw data, which came from different sources, as different researchers were working with different data at the Hospital, and they did not have a unified collection. Thus, the following documentation of the software design will be structured going from the raw data, to the preprocessed data, and then the model itself.

## B.1 Raw Data

The raw data (an example of their contents is found in Figure B.1) was originally scattered amongst many files, registered in different spaces, and do not have masks applied (this, along with other challenges detailed in Section 2.3).

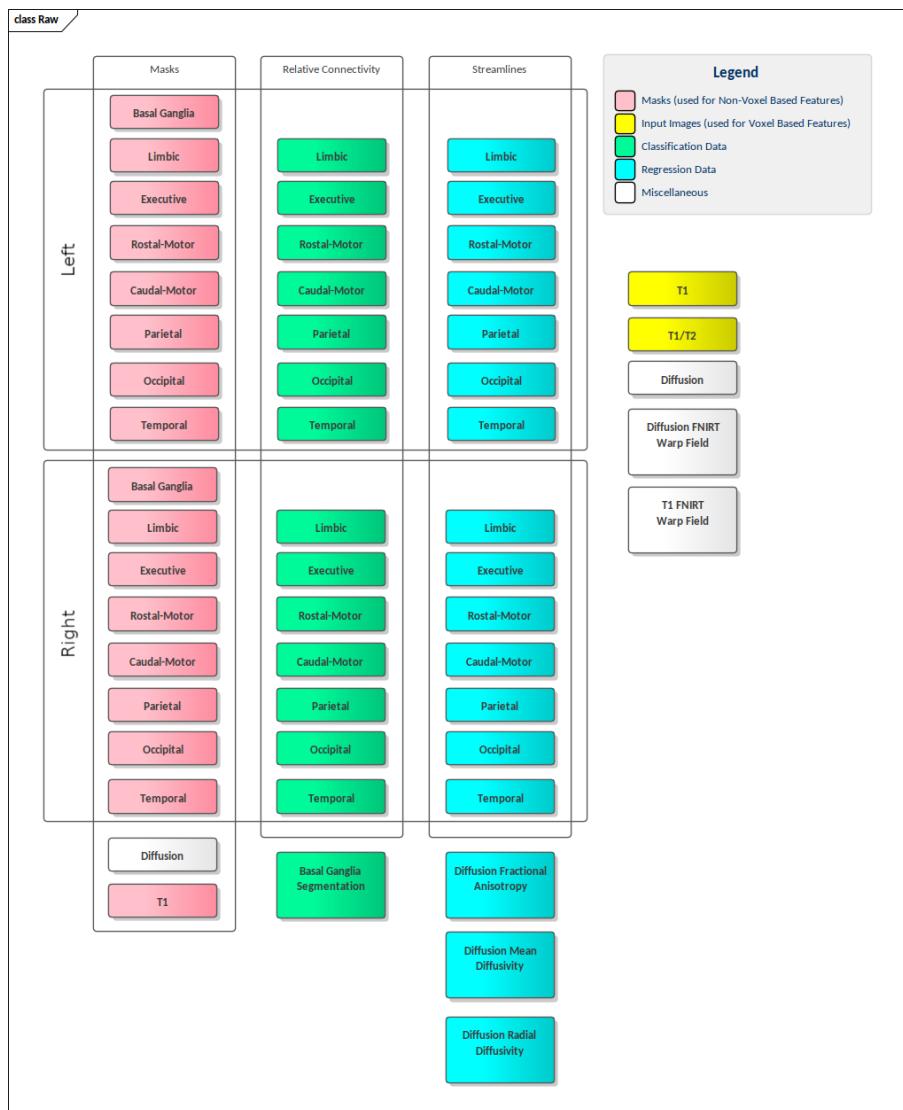


Figure B.1: Files: Raw

The class diagram documentations are detailed abstractions of the actual source code, as unified modeling language (UML) is not completely Python-compliant. Before moving on, the simple data types and primitives seen in Figure B.2 are used in the UML diagrams:

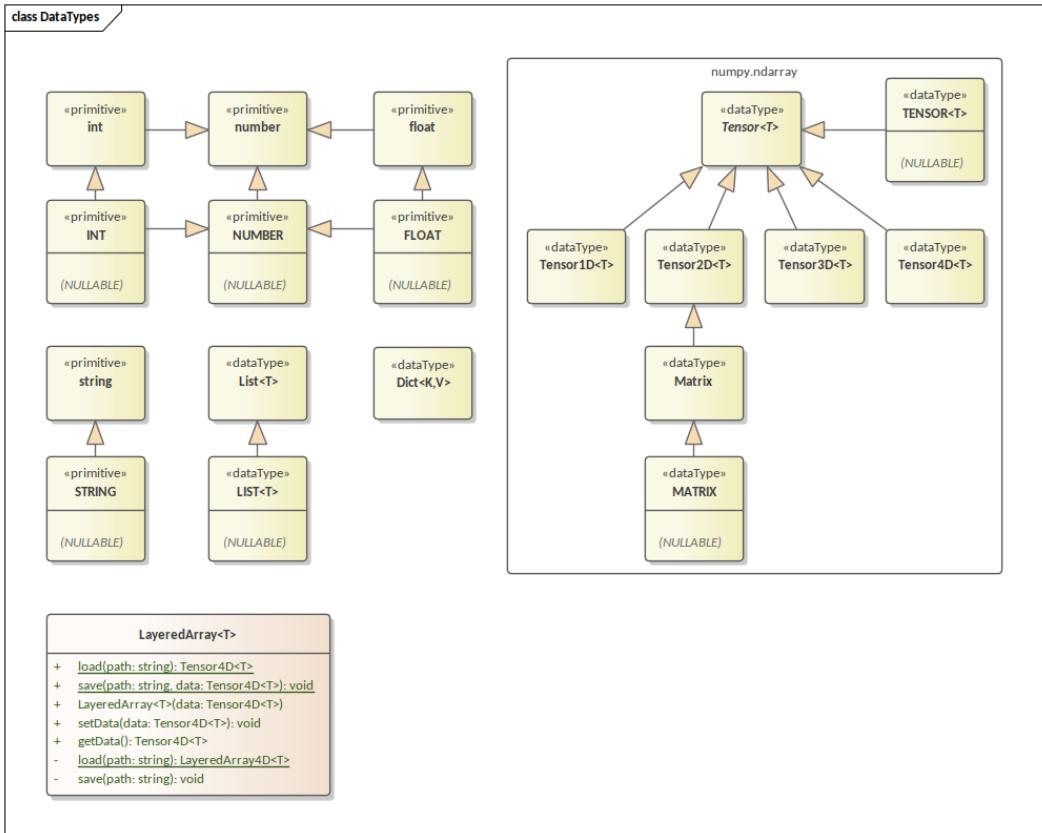


Figure B.2: Class Diagram: Data Types

The odd one out in this diagram is the `LayeredArray`, which is more than a simple data type or primitive. This class is a simple space efficient data structure of a 4D tensor.

Storing the raw tensor of some of the data provided can be highly inefficient. Ultimately, the design aims to collect the scattered data from multiple files into clean logical groups: for example, all the cortical target masks into a single 4D tensor, where the 4th dimension is reserved for the multiple targets. This is a greatly inefficient way of storing the data, though, as each target region only takes up a small portion of the entire space. The idea behind the `LayeredArray` class is that it stores the 4D tensor as a list of 3D tensors, each cropped down to their effective space where there are non-zero voxels. This also requires an additional list of 3D vectors to store each layer's origin, and where to paste it in the original space.

This solution offers a very efficient way of storing data for this use case, as the raw storage solution for the cortical target masks involved around 50MBs, and this solution cuts it down to 3MBs. This is even more drastic for the relative connectivity, which is more than a simple boolean mask and proportionally takes up even less space of the entire brain; in which's case it was cut down from 110MBs to 0.5MBs, reducing the disk requirement by several magnitudes.

The original NIfTI format also does a very good job at storing data efficiently, but this solution provides control over the way of storing the data on a much lower level. This is beneficial as the data are stored in numpy format, making it easier to ignore certain data type safety checks, data

type conversions, and leaving behind the NIfTI format's additional complexity of the orientation, transformation, and many more nuances that are part of the NIfTI header.

This has the undoubted drawback of not being able to use FSL tools natively on our datatypes, such as fsleyes for simply viewing a record. But thanks to the opensource nature of the FSL suite, with a few additional lines of code, support can be added for our datatypes (included in Appendix F).

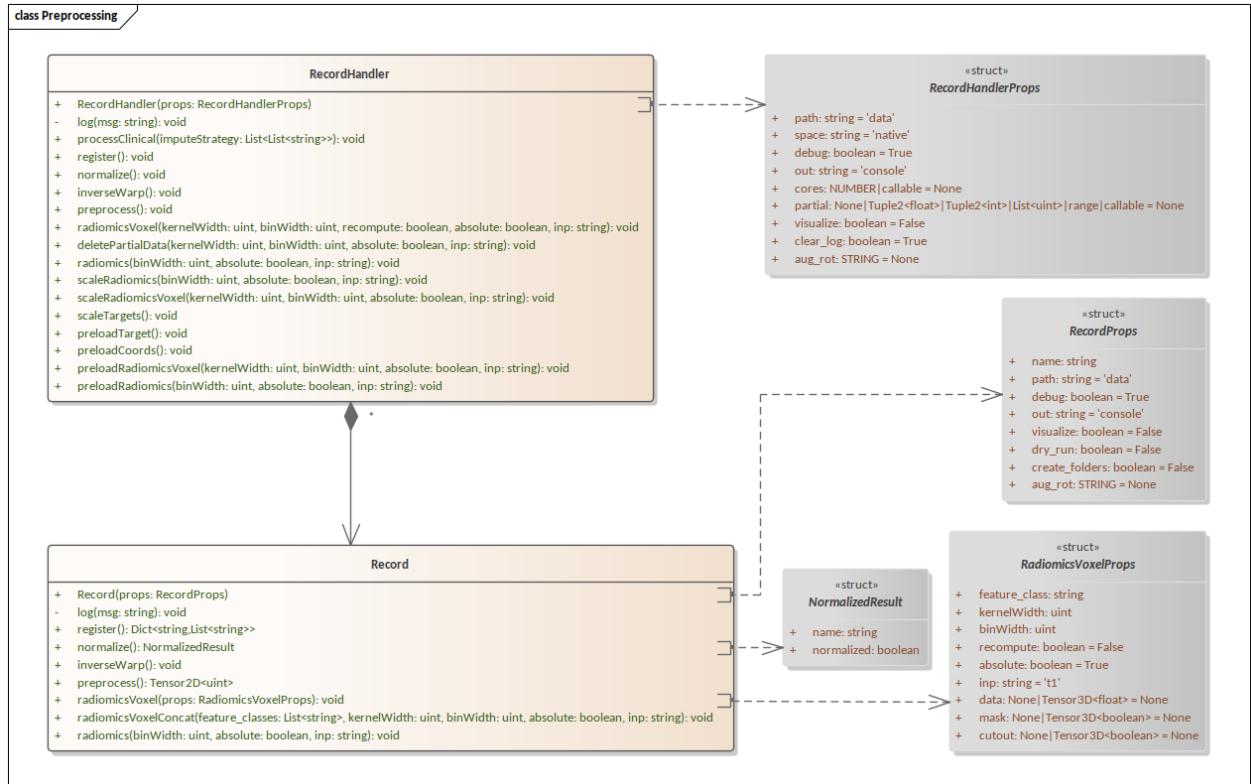


Figure B.3: Class Diagram: Preprocessing

The class diagram shown in Figure B.3 contains the two classes responsible for the preprocessing of the data. The **RecordHandler** being the controller itself that handles the high level operations on a collection of records, and the records handling the low level computations on the data itself.

## B.2 Common Functions

There is a set of static common functions providing the low level backbone of the entire thesis project. They are grouped into two categories, namely util and visual (as seen in Figure B.4), where the former contains everything from simple data type castings, external FSL library system calls, to computing radiomics, and more, and the latter one is a collection of functions for visualizing data.

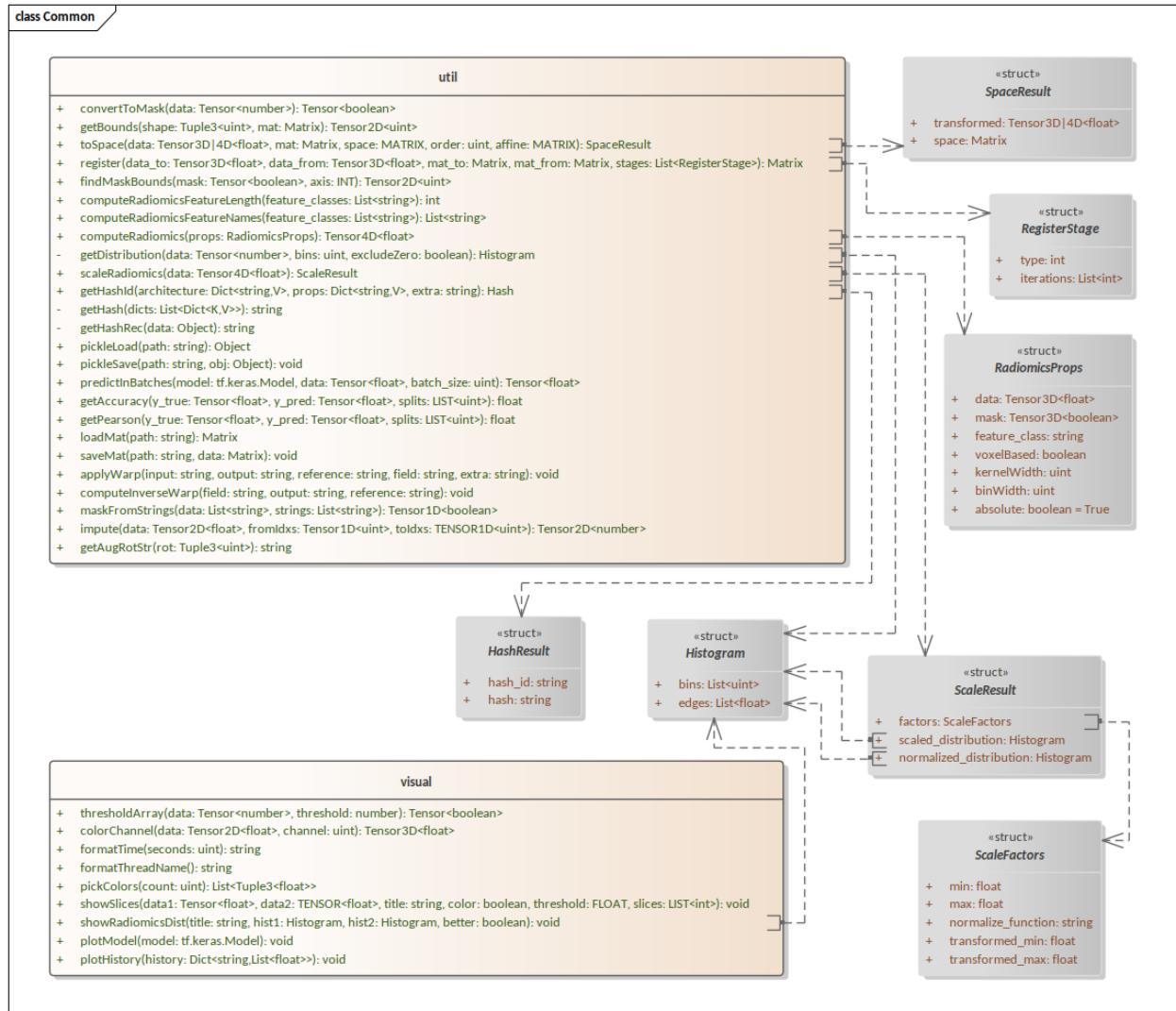


Figure B.4: Class Diagram: Common

### B.3 Preprocessed Data

Preprocessing the data are composed from the next few high level operations done by the Record-Handler:

1. Register some records (not needed for most of them)
2. Compute all normalized records
3. Compute the inverse FNIRT warp field
4. Convert native records into our numpy format (apply the affine transformations, merge different sources, etc.)
5. Compute scaling factors across all native records
6. Preload native records

7. Convert normalized records into our numpy format
8. Compute scaling factors across all normalized records
9. Preload normalized records
10. Construct normalized coordinate maps
11. Warp normalized coordinate maps into native space
12. Scale and preload coordinate maps
13. Impute clinical data

After preprocessing, the set of logical groupings and files seen in Figure B.5 are left, split into native and normalized records:



Figure B.5: Files: Preprocessed

## B.4 Data Generator

As this thesis project focuses only on the voxels inside the Basal Ganglia, this means that the model is never going to operate outside of that volume at a datapoint level.

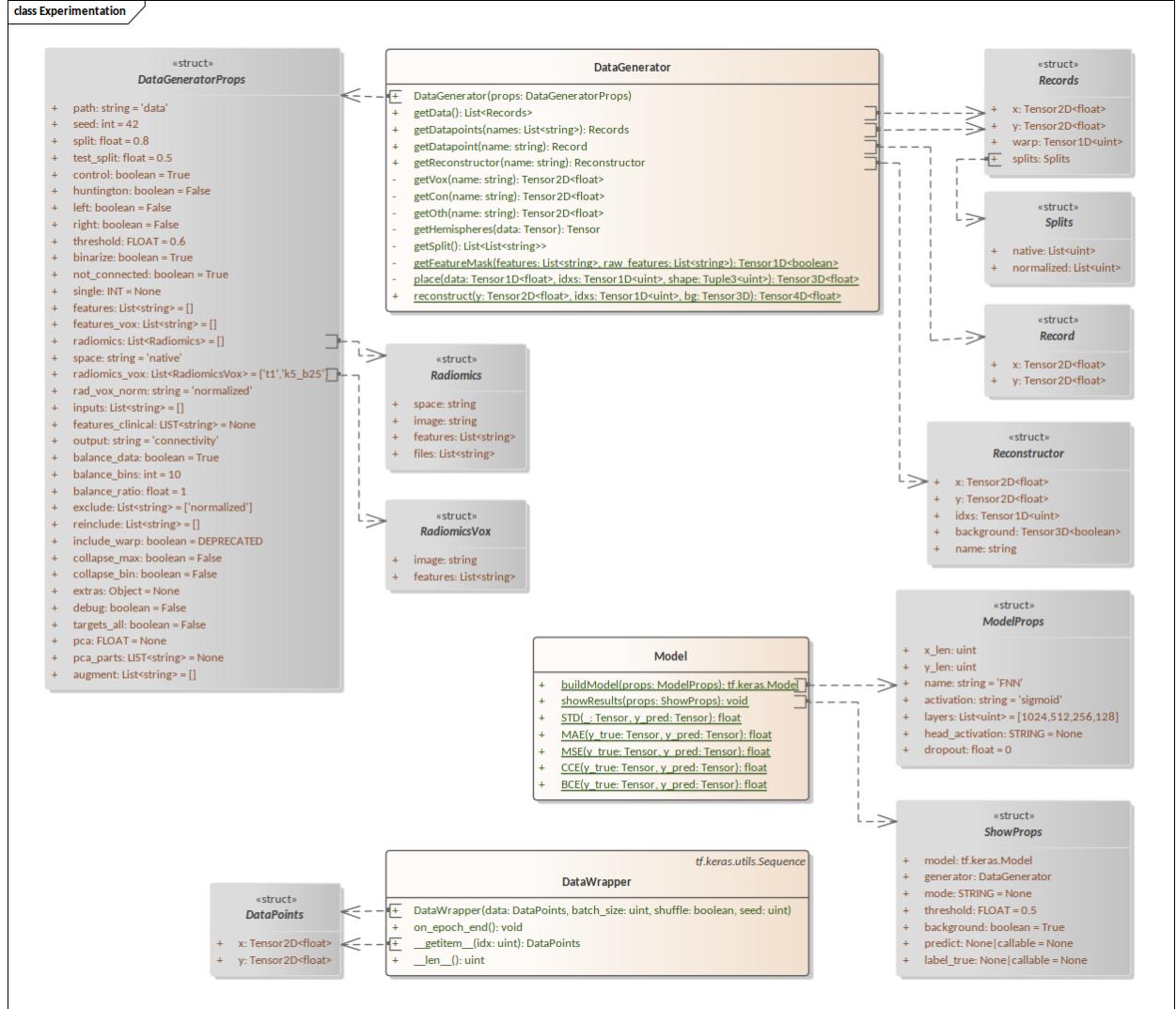


Figure B.6: Class Diagram: Experimentation

The DataGenerator class is responsible for feeding the Model with data, according to the specifications of the experiments (a schematic representation of them are found in Figure B.6). In order to make this more efficient and fast, loading the entire volume of records from disk can be avoided by 'preloading' the ROI from each record. This simply put means that all logical grouping and files specified in Figure B.5 are preloaded into 1D and 2D tensors, where the first dimension is the flattened voxels of the ROI and the second is whatever layers were stored originally in the last dimension of the spatial data (for example the streamlines of the different target regions).

Computing and persisting the preloaded data vastly speeds up the DataGenerator. Although this has the added consequence that the preloaded data yields a pair of tensors instead of a single volume, as there is differentiation between the left and right hemisphere datapoints. And the voxel count in the two hemispheres are not symmetrical, thus the hemisphere differentiation can not be

stored as an extra dimension. A design decision was made to store the left and right hemisphere data in separate files.

The properties of the DataGenerator provided in its constructor, as listed in Table B.1, define the type and format of the generated data.

Value	Type	Description
path	string	path of the data
seed	int	random seed for the train/val/test splits
split	float	train/all ratio
test_split	float	test/(test+validation) ratio
control	boolean	include control records
huntington	boolean	include huntington records
left	boolean	include left hemisphere datapoints
right	boolean	include right hemisphere datapoints
threshold	FLOAT	if not None it thresholds the labels by setting the values under the provided threshold to zero; if 0 it re-one-hot encodes the labels (sets the maximum label to 1 and the rest to 0)
binarize	boolean	if thresholded and True, it also sets the labels above the threshold to 1 (and the rest to 0)
not_connected	boolean	if thresholded and True, it appends an additional 'not connected' label, which complements the sum of the labels per voxel to 1
single	INT	if not None it only returns the label with the provided index
features	List<string>	used non-voxel based radiomics features (emptylist means all)
features_vox	List<string>	used voxel based radiomics features (emptylist means all)
radiomics: List<Radiomics>	space: string	space of non-voxel based radiomic features (native/normalized)
	image: string	input image of non-voxel based radiomic features (t1/t1t2)
	features: List<string>	list of binning settings of the features (b25, b50, b10r... etc.)
	files: List<string>	list of region mask settings of the features (targets/roi/brain)
space	string	space of voxel based radiomic features (native/normalized)
radiomics_vox: List< RadiomicsVox >	image: string	input image of voxel based radiomic features (t1/t1t2)
	features: List<string>	list of binning and kernel settings of the features (k5_b25, k9_b50, k7_b10r... etc.)
rad_vox_norm	string	use normalization, or only min-max scaling on the voxel based radiomic features (norm/scale)
inps	List<string>	additional voxel inputs (t1/t1t2/diffusion/diffusion_fa/ diffusion_md/diffusion_rd)

features_clin	List<string>	additional clinical data inputs (empty array means all)
outp	string	output (connectivity/streamline/basal_seg/diffusion_fa/diffusion_md/diffusion_rd)
balance_data	boolean	enables data balancing
balance_bins	int	number of bins used for continuous data when balancing
balance_ratio	float	ratio of the resampling of the difference between each bin and the max bin when balancing (where 0 is unbalanced and 1 is perfectly balanced)
exclude	List<string>	can manually add missing groups of records to exclude (t1t2/normalized/basal_seg/diffusion_fa)
reinclude	List<string>	can manually re-include (append) missing groups of records to the train split (t1t2/normalized/basal_seg/diffusion_fa)
debug	boolean	only returns 1/1/1 records for train/val/test when True
augment	List<string>	list of record suffixes, used to include augmented records in the training split; for example the suffix '_5_0_0' is used for the rotation augmentation of 5, 0 and 0 degrees on the X, Y and Z axis (naturally these records needs to be computed before using here)

Table B.1: Data Generator Properties

# C Software Implementation

Detailing the entire implementation of the project is beyond the scope of this report. But some solutions contain non self explanatory ideas and are worthy of detailed explanations. For more details please refer to the source code.

## C.1 Interpolation

One crucial detail concerning the application of transformations to the records is the type of interpolation used. For some images like T1, the standard trilinear interpolation makes sense, as the values in-between voxels should be continuous. However, for some other images like binary masks, the interpolation should be the nearest neighbour, as the edge of the mask cannot be interpreted as a fraction. And most importantly, the relative connectivity and streamline images should also be interpolated with the nearest neighbour. In theory, these images could be computed with trilinear interpolation, but this would degrade the images due to their unbalanced nature of having a few voxels with high intensity values relative to the entire voxel space, effectively eroding the boundaries of the high intensity volumes.

In practice this meant using 0th and 1st order spline interpolations, which are numerically the same as nearest neighbour and trilinear interpolations.

## C.2 Multithreading

As most of the preprocessing could not be easily offloaded to the GPU (without re-implementing major libraries such as 'pyradiomics' with GPU support), it was crucial to implement multithreading to save time.

The most straightforward way of doing so was to split the load at a record level, and process multiple records in parallel. Testing revealed an interesting property of the feature extraction, which is the RAM IO operation bottleneck of the process. Extracting voxel-based features happens in a 'voxel batch', which seemed to only increase the RAM usage of the process beyond a certain point, but not the speed. After days of tweaking and running tests, it was concluded that computing the GLCM feature class is RAM IO operation-heavy, meaning that increasing voxel batch size or the number of threads (computing different records in parallel) would not increase the performance past a certain point, as the threads are waiting on RAM IO operations regardless.

The optimal settings with the used Intel Core i5-12600K CPU and 128GBs of 3200MT/s RAM, consisted in using 7 threads, out of which 5 were dedicated to computing the GLCM feature class and the remaining two were for computing other feature classes, with a voxel batch size of 1,000.

Trying to increase performance at this point by increasing the voxel batch size, or the number of threads, only resulted in the same or worse overall performance, due to the computational overhead of splitting and merging the work between more threads, and using a lot more RAM.

### C.3 Warp Pre-Computing

As mentioned in subsection 2.4.2, computing the evaluation metrics both in native and normalized space can be done by reconstructing the spatial records from the predicted datapoints and warping them to native/normalized space, and then re-extracting the datapoints from the warped records. However this would be computationally very expensive to do every time the evaluation metrics are calculated.

The solution is to pre-compute index arrays, by assigning unique indexes to the voxels of the records, warping the records, and mapping the ROI's flattened voxels' unique indexes between the (warped and non-warped) record pairs (as illustrated in Figure C.1) into an index array which can be used to quickly and efficiently to convert the voxels of the ROI between the native and normalized spaces.

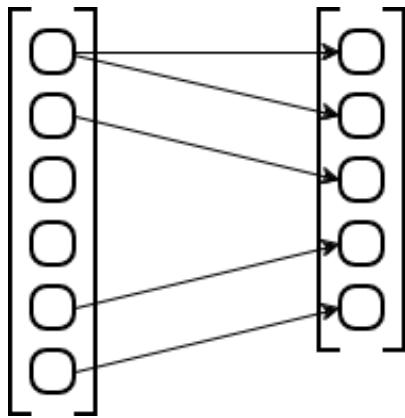


Figure C.1: Index Array Mapping

### C.4 Data Normalization

As noted in subsection 2.3.6, determining which voxel-based radiomic features require log scaling for distribution normalization is handled programmatically. This process involves loading each feature across all records and analyzing their distribution both before and after log scaling.

Initially, the feature is min-max scaled, and the distribution of non-zero voxels is calculated using 100 bins (bin size 0.01). Excluding zero values masks the background, which would distort the distribution. However, this exclusion should have been done with the provided brain masks, which would have been more appropriate since zeros could naturally occur in the features. This oversight, while suboptimal, does not significantly affect the results, as non-background zero counts are negligible. And the only consequence is a potential false negative artifacts (failing to apply normalization where needed), which does not degrade the raw data by inappropriately normalizing features.

Next, the logarithm is applied to the feature prior to min-max scaling, and the distribution is recalculated using the same 100 bin approach. The criteria for selecting normalized features are based on observing an increased standard deviation and a reduced count of voxels in the largest bin after normalization. This ensures that features with 'flattened' and 'stretched' distributions are selected for normalization.

## D Additional Figures

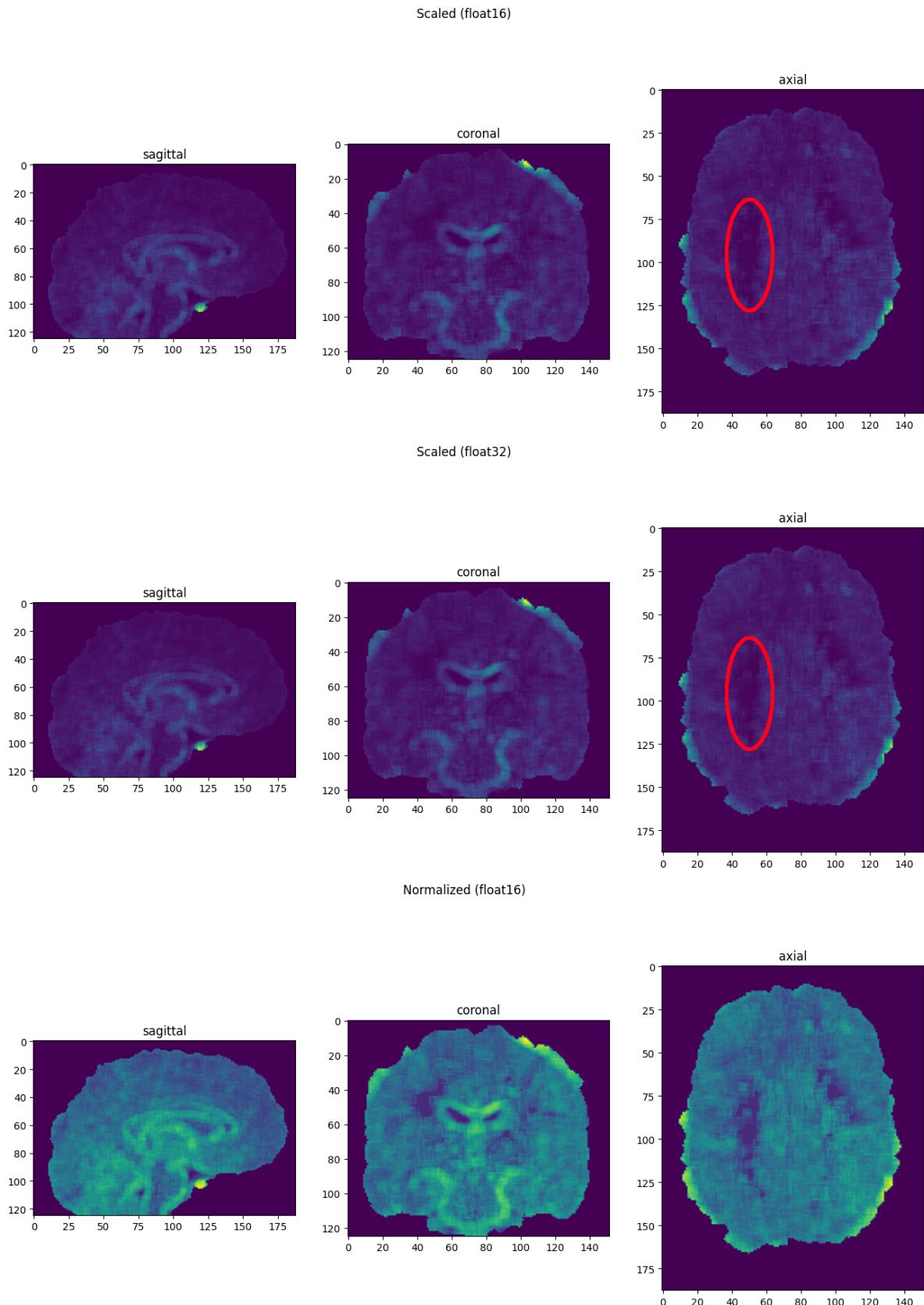


Figure D.1: Loss of Detail in GLDM Small Dependence High Gray Level Emphasis

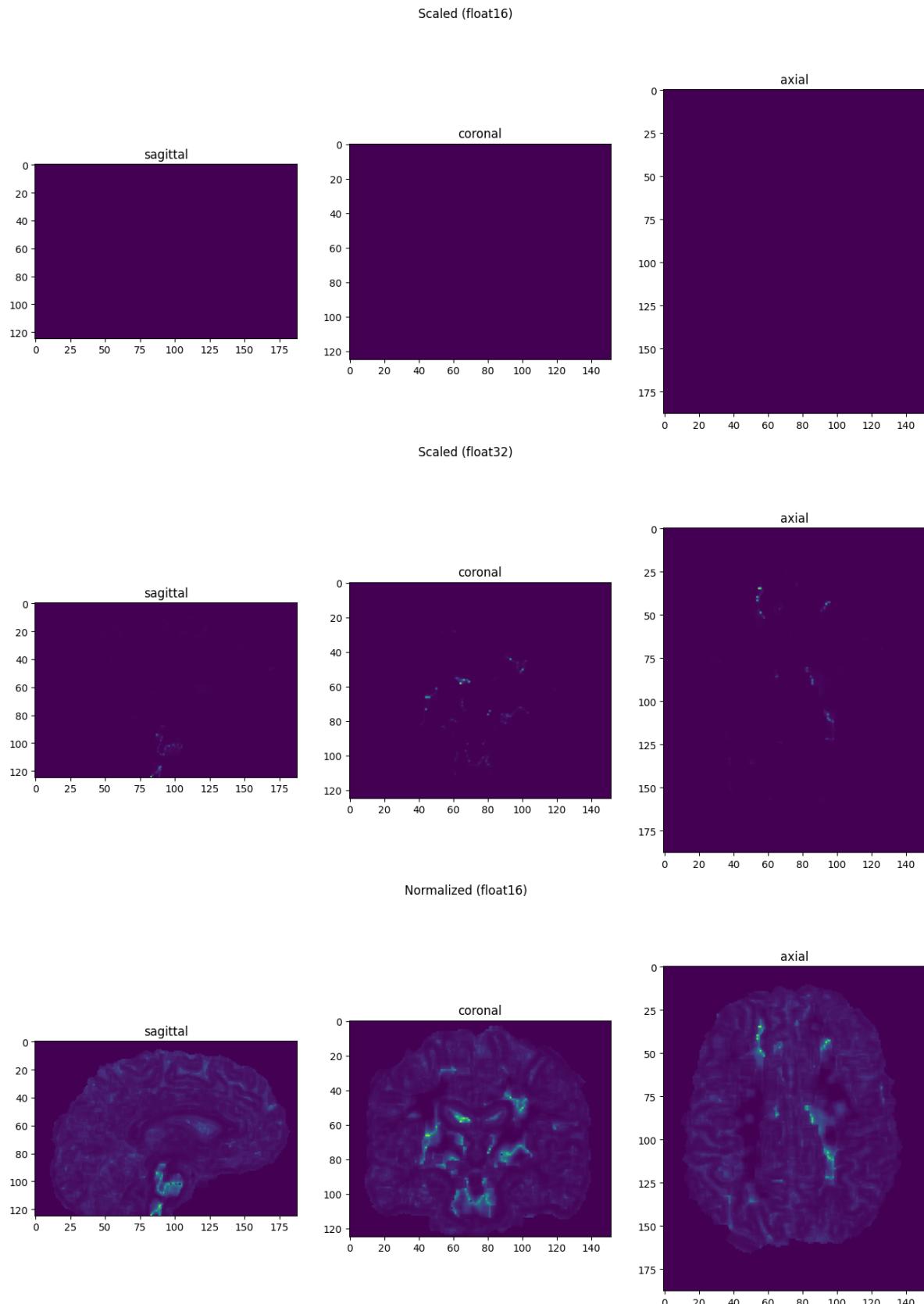


Figure D.2: Loss of Detail in NGTDM Busyness

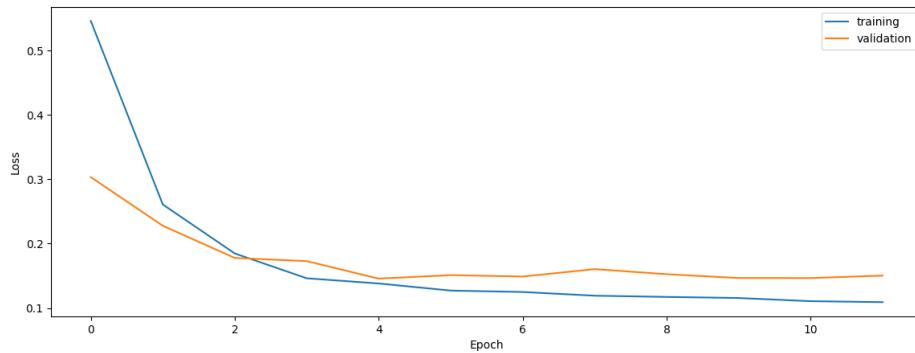


Figure D.3: Training Curve: Subcortical Segmentation Best Performing Model

Color	Region
Red	Caudate
Green	Putamen
Blue	Accumbens

Table D.1: Subcortical Color Coding

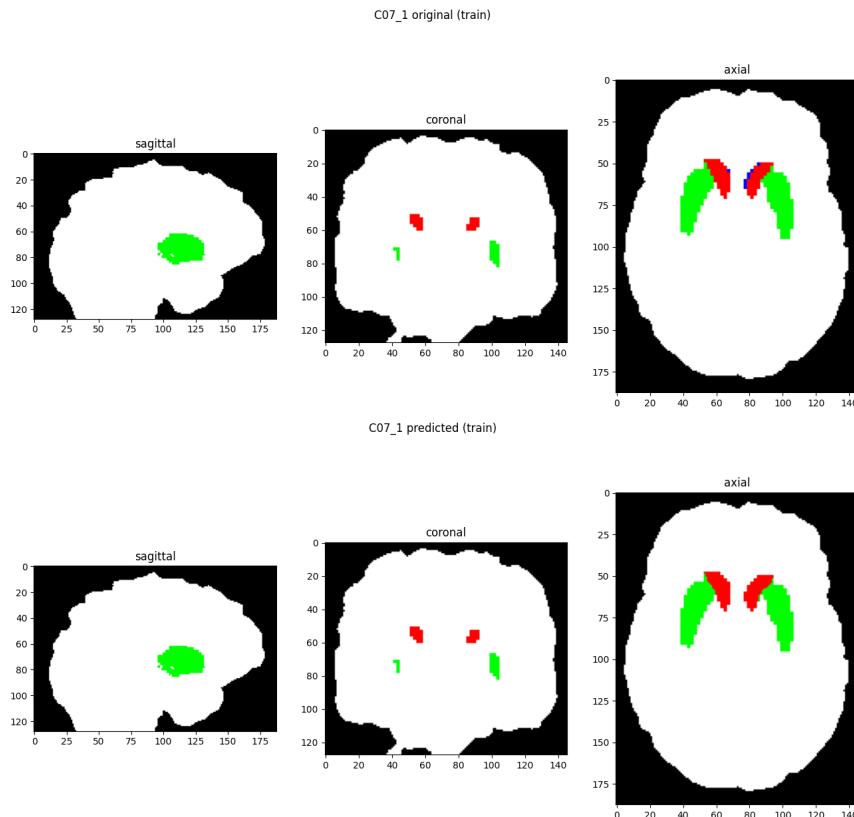


Figure D.4: Train Predictions: Subcortical Segmentation Best Performing Model

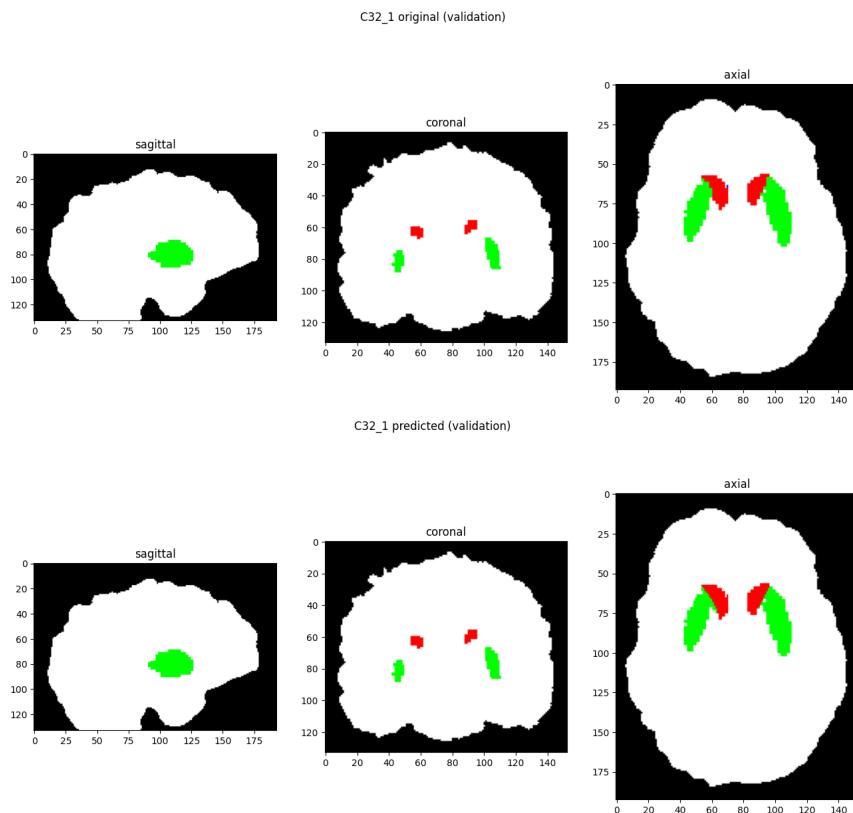


Figure D.5: Validation Predictions: Subcortical Segmentation Best Performing Model

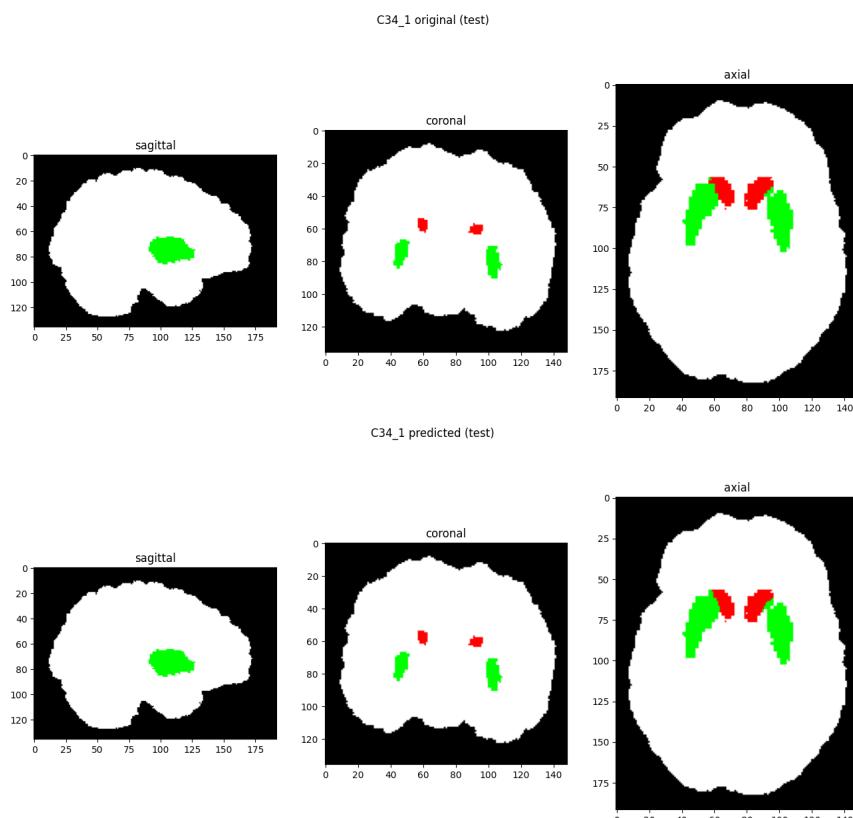


Figure D.6: Test Predictions: Subcortical Segmentation Best Performing Model

## E Additional Tables

First Order	GLCM	GLSZM
Energy	Autocorrelation	SmallAreaEmphasis
TotalEnergy	JointAverage	LargeAreaEmphasis
Entropy	ClusterProminence	GrayLevelNonUniformity
Minimum	ClusterShade	GrayLevelNonUniformityNormalized
10Percentile	ClusterTendency	SizeZoneNonUniformity
90Percentile	Contrast	SizeZoneNonUniformityNormalized
Maximum	Correlation	ZonePercentage
Mean	DifferenceAverage	GrayLevelVariance
Median	DifferenceEntropy	ZoneVariance
InterquartileRange	DifferenceVariance	ZoneEntropy
Range	JointEnergy	LowGrayLevelZoneEmphasis
MeanAbsoluteDeviation	JointEntropy	HighGrayLevelZoneEmphasis
RobustMeanAbsoluteDeviation	Imc1	SmallAreaLowGrayLevelEmphasis
RootMeanSquared	Imc2	SmallAreaHighGrayLevelEmphasis
Skewness	Idm	LargeAreaLowGrayLevelEmphasis
Kurtosis	MCC	LargeAreaHighGrayLevelEmphasis
Variance	Idmn	
Uniformity	Id	
	Idn	
	InverseVariance	
	MaximumProbability	
	SumEntropy	
	SumSquares	

<b>GLRLM</b>	<b>NGTDM</b>	<b>GLDM</b>
ShortRunEmphasis	Coarseness	SmallDependenceEmphasis
LongRunEmphasis	Contrast	LargeDependenceEmphasis
GrayLevelNonUniformity	Busyness	GrayLevelNonUniformity
GrayLevelNonUniformityNormalized	Complexity	DependenceNonUniformity
RunLengthNonUniformity	Strength	DependenceNonUniformityNormalized
RunLengthNonUniformityNormalized		GrayLevelVariance
RunPercentage		DependenceVariance
GrayLevelVariance		DependenceEntropy
RunVariance		LowGrayLevelEmphasis
RunEntropy		HighGrayLevelEmphasis
LowGrayLevelRunEmphasis		SmallDependenceLowGrayLevelEmphasis
HighGrayLevelRunEmphasis		SmallDependenceHighGrayLevelEmphasis
ShortRunLowGrayLevelEmphasis		LargeDependenceLowGrayLevelEmphasis
ShortRunHighGrayLevelEmphasis		LargeDependenceHighGrayLevelEmphasis
LongRunLowGrayLevelEmphasis		
LongRunHighGrayLevelEmphasis		

Table E.1: Voxel Based Radiomic Features

<b>3D Shape</b>
MeshVolume
VoxelVolume
SurfaceArea
SurfaceVolumeRatio
Sphericity
Maximum3DDiameter
Maximum2DDiameterSlice
Maximum2DDiameterColumn
Maximum2DDiameterRow
MajorAxisLength
MinorAxisLength
LeastAxisLength
Elongation
Flatness

Table E.2: Shape Based Radiomic Features

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	single_vox	58.3	64.9	59.7	60.0	64.3	61.1	60.9	65.0	62.5	92
2.	single_vox-single_targets	66.1	67.1	61.4	65.2	69.0	64.6	65.7	69.1	66.1	1576
3.	single_vox-single_roi	62.1	65.4	58.2	62.5	67.2	61.5	63.0	67.3	63.5	304
4.	single_vox-single_brain	66.1	66.9	66.3	65.9	69.4	66.5	66.4	69.3	67.7	198
5.	single_vox-many_brain	64.8	66.7	63.7	64.4	68.4	64.1	64.8	68.4	65.7	474
6.	many_vox-single_brain	78.6	77.7	77.4	78.5	79.4	77.5	78.7	79.0	78.4	934
7.	many_vox-CLR	80.0	79.3	79.6	80.1	81.1	79.5	80.4	80.9	80.0	828
8.	many_vox-CL	79.6	79.4	76.6	79.6	80.8	76.9	80.7	81.1	78.6	828
9.	many_vox-CR	81.9	82.2	79.8	82.3	83.5	79.9	81.9	83.2	79.3	828
10.	many_vox-HLR	78.2	79.0	74.6	77.9	78.6	77.1	78.2	78.7	78.6	828
11.	many_vox-HL	76.7	73.9	73.2	76.1	75.9	74.7	78.2	78.2	78.7	828
12.	many_vox-HR	79.3	80.9	73.7	79.4	80.3	76.6	78.2	78.9	76.6	828
13.	many_vox-HLR-clinical_CAP	80.7	76.4	81.0	80.3	77.1	81.1	80.7	78.3	79.4	829
14.	many_vox-HLR-clinical_UHDRS	76.8	73.3	76.6	76.5	74.3	77.5	76.9	75.6	75.7	832
15.	many_vox-HLR-clinical_all	75.0	66.7	72.9	74.1	72.1	75.3	74.5	73.4	73.4	919
16.	many_vox-CHLR	79.3	80.5	78.5	79.3	81.2	80.0	79.5	81.2	81.0	828
17.	many_vox-CHL	77.5	77.4	74.3	77.5	79.1	76.6	79.0	80.4	79.3	828
18.	many_vox-CHR	81.2	82.7	79.9	81.3	83.3	81.1	80.6	82.6	80.7	828
19.	many_vox-CLR_coords	79.5	79.3	79.0	79.7	81.0	79.0	80.0	80.8	79.6	831
20.	many_vox-CLR_scale	79.6	79.0	78.9	79.8	80.6	78.8	80.0	80.3	79.3	828
21.	many_vox-CLR_b50	80.1	79.5	77.8	80.2	80.2	77.9	80.5	80.1	78.4	828
22.	many_vox-CLR-balance_025	51.6	68.5	73.9	80.3	81.0	79.1	80.5	80.7	79.6	828
23.	many_vox-CLR-balance_050	45.8	66.6	72.7	79.7	80.9	79.1	79.9	80.7	79.6	828
24.	many_vox-CLR-balance_075	41.8	63.5	70.6	80.6	81.0	79.0	80.8	80.8	79.6	828
25.	many_vox-CLR-balance_100	41.1	63.1	67.9	79.1	80.1	78.0	79.3	79.8	78.8	828
27.	many_vox-CLR-reinclude	80.5	79.1	79.3	80.7	81.1	79.3	81.1	80.9	79.7	828

Table E.3: Hyperparameter Tuning: Diffusion Fractional Anisotropy - Native T1

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	single_vox	63.1	70.1	62.2	66.2	69.8	68.2	66.6	69.9	69.2	92
2.	single_vox-single_targets	70.6	71.2	64.3	69.4	71.3	66.5	69.5	71.1	67.6	1576
3.	single_vox-single_roi	67.1	69.0	65.5	66.2	69.8	66.7	66.5	69.8	67.6	304
4.	single_vox-single_brain	67.1	70.6	64.6	67.0	70.8	67.0	67.3	70.7	68.0	198
5.	single_vox-many_brain	67.6	70.3	66.5	67.3	70.8	67.5	67.5	70.8	68.4	474
6.	many_vox-single_brain	81.0	80.7	78.8	80.9	80.5	79.4	81.1	80.5	79.9	934
7.	many_vox-CLR	80.1	79.7	76.3	80.5	79.8	77.8	80.8	79.9	78.7	828
8.	many_vox-CL	77.2	78.1	72.5	77.9	78.5	77.2	79.1	79.5	79.7	828
9.	many_vox-CR	79.9	80.0	77.2	80.6	79.3	78.3	80.1	79.0	78.3	828
10.	many_vox-HLR	79.9	74.3	78.6	79.5	76.2	78.5	79.7	76.1	80.5	828
11.	many_vox-HL	72.1	58.9	71.4	72.6	69.4	71.5	75.3	72.2	76.3	828
12.	many_vox-HR	75.6	75.7	74.9	75.7	76.2	75.0	74.2	74.7	75.0	828
13.	many_vox-HLR-clinical_CAP	75.1	75.2	72.5	75.4	76.0	73.4	75.8	76.8	72.6	829
14.	many_vox-HLR-clinical_UHDRS	80.2	77.4	78.2	79.8	77.7	78.2	80.2	78.5	77.2	832
15.	many_vox-HLR-clinical_all	76.2	70.5	71.5	75.3	73.5	72.9	75.7	74.7	71.8	919
16.	many_vox-CHLR	75.8	74.4	72.6	76.4	75.8	76.4	76.6	75.8	77.7	828
17.	many_vox-CHL	77.4	72.1	72.6	77.6	74.2	77.3	79.2	75.8	80.3	828
18.	many_vox-CHR	75.5	75.6	73.1	76.2	76.0	75.3	75.1	74.8	75.4	828
19.	many_vox-CLR_coords	80.2	80.2	77.4	80.6	80.2	78.4	80.9	80.4	79.3	831
20.	many_vox-CLR_scale	80.8	80.3	75.9	81.2	80.1	77.5	81.5	80.2	78.4	828
21.	many_vox-CLR_b50	79.8	80.6	74.9	80.3	80.2	77.2	80.6	80.1	78.3	828
22.	many_vox-CLR-balance_025	57.0	68.0	74.6	80.8	80.0	78.2	81.1	80.2	79.2	828
23.	many_vox-CLR-balance_050	51.8	64.9	72.6	81.0	80.2	78.3	81.2	80.3	79.2	828
24.	many_vox-CLR-balance_075	47.3	62.6	71.0	80.9	79.8	77.4	81.2	80.0	78.4	828
25.	many_vox-CLR-balance_100	43.8	62.5	70.4	79.6	79.3	77.8	79.9	79.5	78.8	828
28.	many_vox-single_brain-coords	81.4	81.0	79.0	81.3	80.8	79.7	81.6	80.8	80.3	937

Table E.4: Hyperparameter Tuning: Diffusion Fractional Anisotropy - Native T1/T2

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	single_vox	52.5	58.9	53.4	54.2	60.3	54.9	54.4	59.3	54.2	92
2.	single_vox-single_targets	60.1	56.7	58.8	60.3	64.0	61.4	60.5	63.8	60.8	1380
3.	single_vox-single_roi	65.2	67.1	63.3	64.0	68.9	65.6	64.3	68.6	65.2	276
4.	single_vox-single_brain	65.3	66.1	65.8	64.5	68.9	66.0	64.8	68.6	65.9	184
5.	single_vox-many_brain	60.6	62.7	61.8	61.5	66.1	62.5	61.8	65.8	62.0	460
6.	many_vox-single_brain	81.9	82.3	80.8	81.4	83.2	81.0	81.6	83.4	81.5	920
7.	many_vox-CLR	82.3	82.1	80.8	82.0	83.4	81.3	82.3	83.7	81.7	828
8.	many_vox-CL	81.8	81.3	79.8	82.5	83.9	82.9	81.8	83.5	81.6	828
9.	many_vox-CR	83.6	83.2	82.8	82.8	83.5	81.5	83.9	84.2	82.9	828
10.	many_vox-HLR	76.4	72.9	64.0	76.4	75.8	70.0	75.4	74.3	68.5	828
11.	many_vox-HL	77.0	71.8	66.3	79.2	77.9	73.2	76.7	74.4	69.3	828
12.	many_vox-HR	75.0	73.5	60.9	73.0	73.5	66.6	74.1	74.5	67.9	828
13.	many_vox-HLR-clinical_CAP	77.2	77.5	66.1	77.6	78.0	64.1	76.5	76.3	64.0	829
14.	many_vox-HLR-clinical_UHDRS	75.7	75.6	64.6	76.2	76.6	62.8	75.1	74.8	62.6	832
15.	many_vox-HLR-clinical_all	71.8	69.7	65.3	72.4	72.3	65.7	71.0	70.7	64.8	919
16.	many_vox-CHLR	80.8	80.1	75.7	80.6	81.5	79.1	80.3	81.1	78.7	828
17.	many_vox-CHL	79.9	77.7	73.9	81.2	81.9	79.6	79.6	80.0	77.5	828
18.	many_vox-CHR	81.5	82.0	77.5	80.5	81.5	79.0	81.3	82.2	79.9	828
19.	many_vox-CLR_coords	81.7	81.6	81.0	81.5	83.3	81.4	81.7	83.5	81.7	831
20.	many_vox-CLR_scale	84.4	82.9	82.0	84.1	84.4	82.4	84.3	84.8	82.6	828
22.	many_vox-CLR-balance_025	58.2	80.7	85.6	82.7	83.5	81.7	82.9	83.8	82.0	828
23.	many_vox-CLR-balance_050	55.3	79.8	86.5	84.0	84.2	82.4	84.3	84.6	82.7	828
24.	many_vox-CLR-balance_075	51.2	79.2	86.0	83.4	84.0	81.9	83.7	84.4	82.1	828
25.	many_vox-CLR-balance_100	46.2	78.2	84.8	81.9	83.1	81.2	82.1	83.4	81.5	828
27.	many_vox-CLR-reinclude	84.6	83.3	82.3	84.4	84.6	82.8	84.6	84.9	82.9	828
28.	many_vox-scale-bal50-reinclude	96.3	83.8	81.7	76.8	77.1	77.3	76.8	76.6	76.9	828
29.	many_vox-bal50-reinclude	96.2	83.1	83.9	76.9	77.1	77.7	76.9	76.6	77.3	828

Table E.5: Hyperparameter Tuning: Diffusion Fractional Anisotropy - Normalized T1

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	<b>single_vox</b>	58.7	66.4	59.8	61.4	66.7	65.0	61.8	66.2	65.1	92
2.	<b>single_vox-single_targets</b>	59.8	58.0	58.8	58.8	63.9	61.5	59.0	63.8	61.4	1380
3.	<b>single_vox-single_roi</b>	64.3	63.0	66.7	63.0	65.8	66.1	63.3	66.0	66.2	276
4.	<b>single_vox-single_brain</b>	66.6	68.8	67.1	65.8	69.1	67.4	66.1	68.9	67.8	184
5.	<b>single_vox-many_brain</b>	67.3	68.8	66.9	66.3	69.0	67.2	66.5	68.9	67.6	460
6.	<b>many_vox-single_brain</b>	83.4	82.3	82.2	<b>83.1</b>	<b>82.3</b>	<b>82.7</b>	<b>83.2</b>	<b>82.8</b>	<b>82.7</b>	920
7.	<b>many_vox-CLR</b>	80.7	80.9	79.2	80.9	80.3	81.1	80.9	80.7	81.0	828
8.	<b>many_vox-CL</b>	79.1	79.8	73.8	80.4	80.3	81.0	79.5	79.5	79.1	828
9.	<b>many_vox-CR</b>	81.9	81.8	81.9	81.4	80.6	80.8	82.0	81.3	81.8	828
10.	<b>many_vox-HLR</b>	73.9	62.3	66.0	74.5	69.4	68.5	73.6	66.6	67.2	828
11.	<b>many_vox-HL</b>	78.1	58.6	66.3	79.9	70.1	72.1	77.7	65.1	67.3	828
12.	<b>many_vox-HR</b>	74.0	66.5	68.0	72.5	69.0	66.0	73.7	68.2	68.9	828
13.	<b>many_vox-HLR-clinical_CAP</b>	78.0	72.9	69.8	78.6	74.2	70.2	77.7	73.8	68.5	829
14.	<b>many_vox-HLR-clinical_UHDRS</b>	80.3	77.2	71.2	80.6	77.6	71.8	79.7	77.1	70.0	832
15.	<b>many_vox-HLR-clinical_all</b>	69.6	69.0	61.2	70.1	71.2	62.2	68.7	70.0	59.9	919
16.	<b>many_vox-CHLR</b>	79.9	73.1	75.5	79.9	75.7	78.0	79.5	74.6	77.3	828
17.	<b>many_vox-CHL</b>	78.9	70.7	69.7	80.4	75.9	78.1	78.9	73.0	75.2	828
18.	<b>many_vox-CHR</b>	72.1	73.1	69.5	71.4	72.8	69.5	72.7	73.4	72.0	828
19.	<b>many_vox-CLR_coords</b>	83.2	82.6	80.8	83.3	81.6	82.1	83.3	82.2	82.0	831
20.	<b>many_vox-CLR_scale</b>	79.9	80.3	79.1	79.9	79.6	80.8	80.0	80.0	80.8	828
22.	<b>many_vox-CLR-balance_025</b>	68.0	80.2	87.3	81.2	80.3	81.3	81.2	80.9	81.3	828
23.	<b>many_vox-CLR-balance_050</b>	61.6	78.2	87.4	82.0	80.8	81.7	82.1	81.3	81.6	828
24.	<b>many_vox-CLR-balance_075</b>	63.0	78.0	86.5	79.7	79.5	80.5	79.8	79.9	80.6	828
25.	<b>many_vox-CLR-balance_100</b>	59.3	77.4	88.9	83.4	81.4	82.0	83.4	82.0	81.8	828
28.	<b>many_vox-brain-coords-bal50</b>	96.5	82.4	83.6	77.6	79.9	77.1	77.3	80.1	76.7	923
29.	<b>many_vox-brain-coords</b>	83.0	82.5	81.8	82.7	82.4	82.4	82.7	82.9	82.3	923

Table E.6: Hyperparameter Tuning: Diffusion Fractional Anisotropy - Normalized T1/T2

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
27.	<b>normalized-t1-many_vox-CLR-reinclude</b>	84.6	83.3	82.3	84.4	84.6	82.8	84.6	84.9	82.9	828
1.	<b>layers_2048</b>	82.0	82.2	81.1	81.9	83.1	81.3	82.1	83.4	81.7	828
2.	<b>layers_1024</b>	84.1	83.1	82.1	84.0	84.4	82.5	84.1	84.7	82.8	828
3.	<b>layers_512</b>	83.2	83.1	81.5	83.1	84.3	82.2	83.2	84.6	82.5	828
4.	<b>activation_sigmoid</b>	83.4	82.9	81.9	83.3	84.2	82.3	83.5	84.5	82.6	828
5.	<b>activation_relu</b>	85.7	83.1	82.3	85.5	84.0	82.3	85.7	84.3	82.8	828
6.	<b>activation_silu</b>	84.7	83.1	82.7	84.6	84.3	82.4	84.7	84.6	83.0	828
7.	<b>activation_elu</b>	7.2	4.9	7.4	14.3	14.8	14.9	12.5	11.1	13.8	828
8.	<b>batch_10e5</b>	80.2	81.2	80.1	80.3	82.3	80.2	80.4	82.6	80.7	828
9.	<b>batch_10e4</b>	83.1	82.7	81.7	83.0	83.8	82.1	83.2	84.2	82.4	828
10.	<b>batch_10e3</b>	85.5	83.8	82.5	<b>85.3</b>	<b>84.9</b>	<b>82.8</b>	<b>85.5</b>	<b>85.2</b>	<b>83.1</b>	828
11.	<b>batch_10e2</b>	89.3	83.4	83.6	89.1	84.2	84.2	89.3	84.4	84.3	828
12.	<b>dropout_000</b>	89.0	84.8	83.0	88.9	86.0	83.6	88.9	86.0	84.0	828
13.	<b>dropout_025</b>	89.4	84.2	83.9	89.2	85.0	84.0	89.3	85.4	84.4	828
14.	<b>dropout_050</b>	84.6	83.5	82.2	84.4	84.5	82.6	84.6	84.9	82.9	828
15.	<b>dropout_075</b>	89.5	84.2	83.0	89.3	85.7	83.3	89.3	85.9	83.7	828
16.	<b>dropout_090</b>	84.5	83.8	82.4	84.3	84.8	82.7	84.5	85.1	82.9	828
17.	<b>patience_4</b>	85.3	83.4	82.6	85.2	84.9	82.7	85.3	85.1	83.1	828
18.	<b>patience_7</b>	86.3	83.1	82.8	86.2	84.9	83.0	86.3	85.1	83.4	828
19.	<b>patience_10</b>	87.1	84.4	83.3	87.1	85.1	83.8	87.2	85.3	84.2	828
20.	<b>rate_10e-4</b>	82.9	82.7	80.8	82.8	84.0	81.6	82.9	84.3	81.9	828
21.	<b>rate_10e-3</b>	85.2	83.9	82.1	85.1	85.0	82.7	85.3	85.2	83.0	828
22.	<b>rate_10e-2</b>	nan	nan	nan	nan	nan	nan	nan	nan	nan	828

Table E.7: Architecture Tuning: Diffusion Fractional Anisotropy

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	<b>single_vox</b>	92.6	94.3	94.2	93.1	94.6	94.2	93.9	94.9	95.6	92
2.	<b>single_vox-single_targets</b>	93.9	93.6	93.7	93.6	94.4	94.5	94.3	94.6	95.7	1576
3.	<b>single_vox-single_roi</b>	93.7	94.9	92.8	93.6	94.6	94.6	94.3	94.8	95.8	304
4.	<b>single_vox-single_brain</b>	93.4	94.9	94.9	93.6	94.9	94.8	94.3	95.1	96.0	198
5.	<b>single_vox-many_brain</b>	93.2	94.6	94.5	93.4	94.5	94.6	94.1	94.8	95.8	474
6.	<b>many_vox-single_brain</b>	94.7	95.8	94.8	<b>94.7</b>	<b>95.5</b>	<b>95.1</b>	<b>95.4</b>	<b>95.7</b>	<b>96.3</b>	934
7.	<b>many_vox-CLR</b>	94.0	95.3	94.9	94.2	95.2	94.9	94.9	95.4	96.1	828
8.	<b>many_vox-CL</b>	93.8	94.4	94.6	94.2	94.6	94.7	95.7	95.7	96.6	828
9.	<b>many_vox-CR</b>	95.0	96.4	94.7	95.2	96.1	94.9	95.0	95.6	95.0	828
10.	<b>many_vox-HLR</b>	92.4	94.1	93.8	92.4	94.2	93.8	93.3	94.6	94.9	828
11.	<b>many_vox-HL</b>	91.7	94.2	95.8	91.9	94.7	95.5	94.1	96.1	97.3	828
12.	<b>many_vox-HR</b>	91.8	93.7	92.2	91.6	95.1	93.4	91.2	94.3	93.6	828
13.	<b>many_vox-HLR-clinical_CAP</b>	93.3	89.4	95.2	93.2	90.9	95.1	93.9	92.3	95.3	829
14.	<b>many_vox-HLR-clinical_UHDRS</b>	93.6	89.8	94.7	93.4	90.8	94.9	94.1	92.3	95.1	832
15.	<b>many_vox-HLR-clinical_all</b>	93.2	88.6	94.3	92.5	90.4	94.6	93.3	92.0	94.9	919
16.	<b>many_vox-CHLR</b>	93.6	95.1	94.8	93.7	95.4	95.0	94.4	95.7	96.1	828
17.	<b>many_vox-CHL</b>	93.6	94.3	94.4	94.0	95.3	95.4	95.6	96.4	97.0	828
18.	<b>many_vox-CHR</b>	93.7	95.5	93.4	93.9	95.9	94.5	93.5	95.3	94.8	828
19.	<b>many_vox-CLR_coords</b>	94.5	95.7	95.0	94.6	95.5	95.1	95.3	95.7	96.2	831
20.	<b>many_vox-CLR_scale</b>	94.1	95.3	94.9	94.2	95.2	95.0	94.9	95.4	96.1	828
21.	<b>many_vox-CLR_b50</b>	93.8	95.1	94.2	93.9	94.8	94.5	94.8	95.1	95.8	828
22.	<b>many_vox-CLR-balance_025</b>	95.5	95.9	95.8	94.3	95.3	95.0	95.0	95.5	96.1	828
23.	<b>many_vox-CLR-balance_050</b>	95.3	95.8	95.8	94.3	95.2	95.1	95.0	95.4	96.3	828
24.	<b>many_vox-CLR-balance_075</b>	95.2	95.3	95.8	94.0	95.0	94.9	94.8	95.2	96.1	828
25.	<b>many_vox-CLR-balance_100</b>	95.4	95.7	95.8	94.3	95.2	95.0	95.0	95.4	96.2	828
27.	<b>many_vox-CLR-reinclude</b>	94.2	95.2	94.9	94.3	95.0	94.9	95.1	95.3	96.2	828
28.	<b>many_vox-bal25-coords-brain</b>	97.4	96.7	96.7	93.2	94.0	93.6	94.1	94.4	95.1	937
29.	<b>many_vox-single_brain-coords</b>	94.0	95.3	94.9	94.2	95.1	95.0	94.9	95.4	96.2	937

Table E.8: Hyperparameter Tuning: Mean Diffusivity - Native T1

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	<b>single_vox</b>	93.0	93.8	94.1	93.6	94.7	94.3	94.3	95.0	95.6	92
2.	<b>single_vox-single_targets</b>	94.4	93.9	91.7	93.9	94.7	93.8	94.4	94.9	95.0	1576
3.	<b>single_vox-single_roi</b>	94.2	95.2	94.3	93.8	94.8	94.6	94.5	95.1	95.7	304
4.	<b>single_vox-single_brain</b>	93.9	95.5	92.2	93.8	94.9	94.0	94.4	95.1	95.2	198
5.	<b>single_vox-many_brain</b>	94.0	95.4	91.7	93.7	94.8	93.8	94.3	95.0	95.1	474
6.	<b>many_vox-single_brain</b>	94.1	95.2	93.8	94.1	95.0	94.5	94.7	95.3	95.7	934
7.	<b>many_vox-CLR</b>	94.9	95.1	94.4	<b>95.1</b>	<b>95.8</b>	<b>94.8</b>	<b>95.6</b>	<b>96.1</b>	<b>95.6</b>	828
8.	<b>many_vox-CL</b>	94.1	93.2	95.5	94.6	95.3	95.4	95.9	96.4	96.9	828
9.	<b>many_vox-CR</b>	95.0	96.3	91.4	95.3	96.6	92.8	94.9	96.1	92.0	828
10.	<b>many_vox-HLR</b>	92.2	94.6	95.5	92.6	94.7	95.3	93.3	94.9	96.4	828
11.	<b>many_vox-HL</b>	92.9	94.0	96.1	93.5	94.5	95.9	95.3	96.0	97.6	828
12.	<b>many_vox-HR</b>	92.7	94.9	94.6	93.3	95.1	95.5	92.8	94.2	95.4	828
13.	<b>many_vox-HLR-clinical_CAP</b>	92.7	90.5	93.3	92.9	91.9	94.2	93.6	93.0	94.3	829
14.	<b>many_vox-HLR-clinical_UHDRS</b>	93.0	90.6	93.5	92.8	91.8	94.1	93.5	93.0	94.2	832
15.	<b>many_vox-HLR-clinical_all</b>	92.8	88.9	93.9	92.2	90.9	94.3	93.1	92.1	94.5	919
16.	<b>many_vox-CHLR</b>	93.3	94.7	94.9	93.7	95.3	95.0	94.4	95.6	96.1	828
17.	<b>many_vox-CHL</b>	93.0	92.8	95.6	93.7	95.0	95.6	95.3	96.3	97.1	828
18.	<b>many_vox-CHR</b>	93.9	96.2	94.5	94.5	96.4	94.9	94.1	95.8	94.7	828
19.	<b>many_vox-CLR_coords</b>	94.8	95.2	94.3	95.0	95.9	94.6	95.5	96.2	95.6	831
20.	<b>many_vox-CLR_scale</b>	94.4	94.9	94.3	94.7	95.8	94.5	95.3	96.1	95.4	828
21.	<b>many_vox-CLR_b50</b>	94.7	95.1	94.5	94.9	95.8	94.6	95.4	96.1	95.6	828
22.	<b>many_vox-CLR-balance_025</b>	96.4	95.4	95.7	94.7	95.8	94.6	95.2	96.1	95.6	828
23.	<b>many_vox-CLR-balance_050</b>	96.6	95.5	95.1	94.9	95.8	94.7	95.5	96.1	95.6	828
24.	<b>many_vox-CLR-balance_075</b>	96.3	95.2	95.4	94.6	95.7	94.6	95.2	96.0	95.6	828
25.	<b>many_vox-CLR-balance_100</b>	96.3	95.3	95.2	94.7	95.7	94.9	95.3	96.0	95.9	828

Table E.9: Hyperparameter Tuning: Mean Diffusivity - Native T1/T2

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	<b>single_vox</b>	87.6	88.6	88.7	88.7	91.7	91.5	87.8	90.8	90.0	92
2.	<b>single_vox-single_targets</b>	90.1	89.7	91.2	90.1	92.3	92.9	89.3	91.5	91.5	1380
3.	<b>single_vox-single_roi</b>	90.5	91.3	88.7	90.6	92.6	93.0	89.7	91.7	91.5	276
4.	<b>single_vox-single_brain</b>	89.8	90.2	89.9	90.3	92.5	93.1	89.5	91.6	91.7	184
5.	<b>single_vox-many_roi</b>	90.4	90.8	89.4	90.4	92.5	92.9	89.5	91.7	91.4	828
6.	<b>many_vox-single_roi</b>	94.3	93.7	92.0	94.2	95.0	94.8	93.8	94.8	93.9	1012
7.	<b>many_vox-CLR</b>	93.9	93.6	92.5	94.1	95.0	95.1	93.8	94.7	94.3	828
8.	<b>many_vox-CL</b>	94.3	92.3	93.1	95.4	95.2	96.2	94.4	94.1	95.2	828
9.	<b>many_vox-CR</b>	92.2	94.1	91.4	90.4	93.9	91.6	92.2	94.8	92.5	828
10.	<b>many_vox-HLR</b>	92.0	89.2	86.9	92.9	90.9	90.8	91.6	89.7	88.1	828
11.	<b>many_vox-HL</b>	92.1	90.0	88.2	93.8	91.7	92.3	91.9	89.7	89.0	828
12.	<b>many_vox-HR</b>	87.7	85.3	81.0	85.4	84.8	84.4	87.0	86.3	85.9	828
13.	<b>many_vox-HLR-clinical_CAP</b>	90.0	90.2	78.1	91.1	92.0	85.1	89.5	90.8	82.3	829
14.	<b>many_vox-HLR-clinical_UHDRS</b>	82.9	87.5	77.7	86.1	88.2	85.7	83.6	86.7	83.1	832
15.	<b>many_vox-HLR-clinical_all</b>	84.8	83.3	80.7	86.5	88.0	86.3	84.1	86.5	83.8	919
16.	<b>many_vox-CHLR</b>	91.8	91.9	90.3	92.7	92.6	93.2	91.7	91.9	91.6	828
17.	<b>many_vox-CHL</b>	93.1	92.2	91.6	94.7	93.9	94.8	93.3	92.5	92.9	828
18.	<b>many_vox-CHR</b>	91.1	92.1	89.5	89.4	90.8	89.6	91.0	91.9	90.4	828
19.	<b>many_vox-CLR_coords</b>	94.0	93.8	93.2	94.2	95.1	95.3	93.9	94.9	94.5	831
20.	<b>many_vox-CLR_scale</b>	93.7	93.5	92.4	94.0	94.9	94.9	93.6	94.6	94.0	828
22.	<b>many_vox-CLR-balance_025</b>	95.2	95.0	94.6	94.1	95.0	95.1	93.7	94.7	94.3	828
23.	<b>many_vox-CLR-balance_050</b>	95.0	94.5	94.2	94.1	95.0	95.2	93.7	94.7	94.4	828
24.	<b>many_vox-CLR-balance_075</b>	95.0	94.3	94.6	94.2	95.0	95.2	93.8	94.7	94.4	828
25.	<b>many_vox-CLR-balance_100</b>	95.4	94.3	94.5	94.3	95.0	95.1	93.9	94.7	94.3	828
27.	<b>many_vox-CLR-reinclude</b>	93.5	93.3	93.6	93.9	94.8	95.2	93.4	94.6	94.5	828
28.	<b>many_vox-single_roi-coords</b>	94.8	93.8	92.6	<b>94.6</b>	<b>95.2</b>	<b>95.1</b>	<b>94.2</b>	<b>94.9</b>	<b>94.3</b>	1015

Table E.10: Hyperparameter Tuning: Mean Diffusivity - Normalized T1

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	single_vox	86.5	89.4	90.2	88.9	90.8	92.6	88.2	90.7	91.7	92
2.	single_vox-single_targets	90.5	91.1	91.9	90.1	91.8	93.3	89.5	91.7	92.3	1380
3.	single_vox-single_roi	90.5	91.3	90.5	90.4	91.8	93.5	89.8	91.7	92.5	276
4.	single_vox-single_brain	91.1	91.3	92.0	90.9	92.0	93.6	90.3	92.0	92.6	184
5.	single_vox-many_brain	90.7	91.4	92.6	90.5	91.9	93.7	89.9	91.8	92.7	460
6.	many_vox-single_brain	93.5	93.6	94.1	93.7	94.7	94.8	93.3	94.7	94.1	920
7.	many_vox-CLR	92.4	91.2	93.5	93.0	93.8	94.4	92.6	93.9	93.8	828
8.	many_vox-CL	90.1	90.2	92.1	92.4	93.2	94.4	91.0	92.2	93.3	828
9.	many_vox-CR	91.3	91.9	92.5	90.0	93.2	90.6	91.9	94.4	92.1	828
10.	many_vox-HLR	90.0	84.2	89.0	91.4	86.1	91.4	90.1	84.6	89.7	828
11.	many_vox-HL	90.2	84.0	90.5	92.8	88.1	93.2	91.2	85.3	90.9	828
12.	many_vox-HR	86.3	81.8	86.3	85.0	79.6	86.7	86.5	81.8	88.3	828
13.	many_vox-HLR-clinical_CAP	87.1	91.1	84.9	89.7	91.6	88.3	88.0	90.7	86.6	829
14.	many_vox-HLR-clinical_UHDRS	83.0	88.7	76.7	87.3	89.6	85.1	85.3	88.5	82.6	832
15.	many_vox-HLR-clinical_all	88.2	89.2	81.3	89.1	91.5	87.6	87.5	90.6	86.0	919
16.	many_vox-CHLR	91.7	88.8	90.7	92.8	90.6	93.1	92.0	89.7	92.0	828
17.	many_vox-CHL	91.7	88.7	91.6	93.7	91.6	94.5	92.4	89.8	92.8	828
18.	many_vox-CHR	90.6	88.7	89.7	89.1	87.2	89.0	90.7	88.6	90.4	828
19.	many_vox-CLR_coords	93.3	91.1	93.9	93.7	94.2	94.7	93.4	94.2	94.1	831
20.	many_vox-CLR_scale	92.5	91.3	93.4	93.1	93.9	94.4	92.7	94.0	93.8	828
22.	many_vox-CLR-balance_025	93.6	91.7	95.2	93.3	93.9	94.6	92.9	94.0	94.0	828
23.	many_vox-CLR-balance_050	93.4	91.3	94.9	93.3	94.0	94.5	92.9	94.0	93.9	828
24.	many_vox-CLR-balance_075	93.2	90.7	94.9	93.3	93.9	94.5	93.0	94.0	93.9	828
25.	many_vox-CLR-balance_100	93.2	90.4	94.8	93.4	94.0	94.5	93.0	94.0	93.9	828
28.	many_vox-bal100-coords-brain	96.9	95.8	93.0	92.3	93.8	93.3	91.4	93.3	92.2	923
29.	many_vox-single_brain-coords	93.7	93.7	94.2	93.8	94.8	94.9	93.5	94.8	94.3	923

Table E.11: Hyperparameter Tuning: Mean Diffusivity - Normalized T1/T2

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
6.	<b>native-t1-many_vox-single_brain</b>	94.7	95.8	94.8	94.7	95.5	95.1	95.4	95.7	96.3	934
1.	<b>layers_2048</b>	94.6	95.8	95.1	94.6	95.5	95.2	95.3	95.7	96.4	934
2.	<b>layers_1024</b>	94.2	95.5	95.0	94.3	95.2	95.0	95.0	95.4	96.2	934
3.	<b>layers_512</b>	94.3	95.4	94.9	94.4	95.2	95.0	95.1	95.4	96.2	934
4.	<b>activation_sigmoid</b>	94.2	95.4	95.0	94.3	95.2	95.0	95.0	95.4	96.2	934
5.	<b>activation_relu</b>	94.6	95.5	94.9	94.6	95.2	95.0	95.3	95.4	96.2	934
6.	<b>activation_silu</b>	95.3	95.8	94.5	95.1	95.4	94.9	95.7	95.5	96.0	934
8.	<b>batch_10e5</b>	93.0	94.6	94.3	93.3	94.3	94.3	94.1	94.6	95.8	934
9.	<b>batch_10e4</b>	94.0	95.4	94.9	94.1	95.1	94.9	94.8	95.3	96.1	934
10.	<b>batch_10e3</b>	95.1	95.9	94.7	<b>95.0</b>	<b>95.5</b>	<b>95.1</b>	<b>95.6</b>	<b>95.7</b>	<b>96.2</b>	934
11.	<b>batch_10e2</b>	95.7	95.8	94.5	95.5	95.3	94.9	96.0	95.4	95.8	934
12.	<b>dropout_000</b>	94.9	95.8	94.8	94.9	95.5	95.1	95.5	95.6	96.2	934
13.	<b>dropout_025</b>	95.6	95.9	94.5	95.5	95.4	94.9	96.0	95.5	96.0	934
14.	<b>dropout_050</b>	94.9	95.9	94.7	94.9	95.6	95.0	95.5	95.7	96.2	934
15.	<b>dropout_075</b>	94.5	95.7	95.0	94.5	95.4	95.2	95.2	95.6	96.3	934
16.	<b>dropout_090</b>	94.7	95.8	94.7	94.7	95.4	95.0	95.4	95.6	96.2	934
17.	<b>patience_4</b>	94.2	95.5	95.0	94.3	95.2	95.0	95.0	95.4	96.2	934
18.	<b>patience_7</b>	95.2	95.9	94.7	95.1	95.5	95.1	95.7	95.6	96.1	934
19.	<b>patience_10</b>	94.6	95.9	94.9	94.6	95.5	95.1	95.3	95.7	96.2	934
20.	<b>rate_10e-4</b>	94.4	95.6	94.9	94.5	95.3	95.2	95.2	95.5	96.3	934
21.	<b>rate_10e-3</b>	94.9	95.9	94.8	94.8	95.5	95.2	95.5	95.6	96.3	934
22.	<b>rate_10e-2</b>	nan	nan	nan	nan	nan	nan	nan	nan	nan	934

Table E.12: Architecture Tuning: Mean Diffusivity

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	<b>single_vox</b>	63.7	66.6	66.4	63.8	66.3	66.8	63.9	66.3	67.0	92
2.	<b>single_vox-single_targets</b>	65.4	65.6	62.1	65.5	65.3	62.8	65.3	65.0	63.2	1576
3.	<b>single_vox-single_roi</b>	64.9	65.9	61.2	65.0	65.5	61.9	64.7	65.2	62.3	304
4.	<b>single_vox-single_brain</b>	64.2	66.1	65.6	64.4	65.8	66.0	64.0	65.6	66.3	198
5.	<b>single_vox-many_brain</b>	64.2	66.0	65.9	64.3	65.6	66.3	64.2	65.5	66.7	474
6.	<b>many_vox-single_targets</b>	72.6	70.1	67.2	72.7	70.1	67.8	72.2	69.1	68.0	2312
7.	<b>many_vox-single_roi</b>	71.1	71.6	67.6	71.1	71.3	68.0	70.5	70.8	67.7	1040
8.	<b>many_vox-single_brain</b>	72.0	72.2	70.9	72.1	72.0	71.0	71.6	71.2	71.0	934
9.	<b>many_vox-CLR</b>	72.1	72.2	71.4	<b>72.2</b>	<b>71.9</b>	<b>71.5</b>	<b>71.8</b>	<b>71.3</b>	<b>71.5</b>	828
10.	<b>many_vox-CL</b>	71.6	69.3	71.3	71.7	68.7	71.4	71.4	68.1	72.2	828
11.	<b>many_vox-CR</b>	73.9	73.9	69.1	73.9	73.9	69.1	73.6	73.3	68.2	828
12.	<b>many_vox-HLR</b>	70.0	67.6	66.1	69.9	67.9	66.9	69.8	67.1	66.7	828
13.	<b>many_vox-HL</b>	65.8	63.9	66.3	65.6	63.8	67.3	65.2	63.1	68.2	828
14.	<b>many_vox-HR</b>	70.4	67.6	68.3	70.3	68.0	68.5	70.1	66.7	67.2	828
15.	<b>many_vox-HLR-clinical_CAP</b>	70.3	68.8	64.6	70.3	69.0	64.6	70.0	69.6	64.4	829
16.	<b>many_vox-HLR-clinical_UHDRS</b>	69.7	68.1	67.1	69.7	68.5	67.1	69.3	68.8	67.2	832
17.	<b>many_vox-HLR-clinical_all</b>	69.6	65.5	60.4	69.7	66.0	60.4	69.3	66.3	60.2	919
18.	<b>many_vox-CHLR</b>	71.6	70.7	70.4	71.5	70.5	70.2	71.2	70.1	70.2	828
19.	<b>many_vox-CHL</b>	70.9	69.0	70.5	70.8	68.8	70.8	70.6	69.1	71.9	828
20.	<b>many_vox-CHR</b>	70.6	71.7	69.4	70.4	71.7	69.6	70.3	71.1	69.0	828
21.	<b>many_vox-CLR_coords</b>	72.7	72.5	70.8	72.7	72.3	70.8	72.4	71.8	71.0	831
22.	<b>many_vox-CLR_scale</b>	72.0	71.7	69.8	72.0	71.4	69.7	71.5	70.4	69.5	828
23.	<b>many_vox-CLR_b50</b>	70.6	71.0	71.2	70.6	70.8	71.2	70.4	70.6	71.7	828
24.	<b>many_vox-feature_10</b>	71.7	71.7	70.8	71.8	71.5	70.8	71.4	70.9	70.7	738
25.	<b>many_vox-feature_20</b>	71.7	72.1	71.2	71.8	71.9	71.1	71.4	71.1	71.1	648
26.	<b>many_vox-feature_30</b>	71.6	72.1	70.4	71.7	71.9	70.4	71.5	71.4	70.1	558
27.	<b>many_vox-feature_40</b>	71.9	71.9	69.9	72.0	71.7	69.9	71.7	71.2	69.9	468
28.	<b>many_vox-balance_025</b>	63.1	60.8	61.4	65.2	65.6	61.0	65.0	64.9	62.1	828
29.	<b>many_vox-balance_050</b>	67.0	61.5	61.5	60.3	59.8	53.6	60.2	58.8	54.5	828
30.	<b>many_vox-balance_075</b>	69.3	61.7	62.6	59.7	59.4	51.3	59.8	58.2	52.4	828
31.	<b>many_vox-balance_100</b>	64.9	61.3	62.0	52.3	53.7	46.8	52.4	53.2	48.0	828
32.	<b>many_vox-reinclude</b>	72.6	71.9	70.6	72.6	71.6	70.6	72.2	71.0	70.6	828
33.	<b>many_vox-CLR-aug</b>	72.8	72.6	70.3	72.9	72.3	70.3	72.5	71.7	70.6	828

Table E.13: Hyperparameter Tuning: Relative Connectivity - Native T1

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	single_vox	62.7	65.8	66.1	62.8	65.4	66.4	62.7	65.3	66.4	92
2.	single_vox-single_targets	65.0	66.1	61.4	65.2	65.8	61.7	64.9	65.3	61.8	1576
3.	single_vox-single_roi	65.8	66.8	60.3	65.9	66.5	60.4	65.6	66.1	60.2	304
4.	single_vox-single_brain	63.9	66.0	65.6	64.0	65.7	65.7	63.8	65.5	65.5	198
5.	single_vox-many_brain	64.0	65.4	62.7	64.1	65.2	62.7	63.8	64.9	62.3	474
6.	many_vox-single_targets	72.3	71.4	69.7	72.4	71.2	69.9	72.1	70.6	69.8	2312
7.	many_vox-single_roi	73.2	71.6	70.0	73.2	71.5	70.2	72.9	71.0	70.1	1040
8.	many_vox-single_brain	72.2	71.8	71.4	72.3	71.7	71.3	71.9	71.1	71.1	934
9.	many_vox-CLR	70.8	71.8	69.8	70.9	71.5	70.0	70.6	71.4	69.7	828
10.	many_vox-CL	70.4	70.1	68.0	70.5	69.7	68.2	70.1	70.4	68.5	828
11.	many_vox-CR	70.7	72.4	68.6	70.8	72.3	68.7	70.5	71.9	68.5	828
12.	many_vox-HLR	69.0	61.7	68.4	69.0	62.4	68.7	68.7	62.1	68.7	828
13.	many_vox-HL	67.9	64.0	68.3	67.9	64.2	69.6	67.8	64.0	70.6	828
14.	many_vox-HR	68.6	63.0	69.8	68.7	64.0	69.8	68.2	63.5	70.1	828
15.	many_vox-HLR-clinical_CAP	70.3	68.7	65.6	70.3	68.8	65.6	70.0	69.2	65.1	829
16.	many_vox-HLR-clinical_UHDRS	70.6	67.7	66.0	70.6	68.0	66.0	70.3	68.5	65.6	832
17.	many_vox-HLR-clinical_all	69.5	65.0	59.7	69.6	65.3	59.8	69.1	65.6	59.8	919
18.	many_vox-CHLR	70.0	69.2	70.2	69.9	69.1	70.6	69.6	68.6	70.2	828
19.	many_vox-CHL	71.4	68.4	70.1	71.3	68.0	70.9	71.0	67.5	71.0	828
20.	many_vox-CHR	70.5	69.5	71.2	70.4	69.6	71.2	70.2	69.7	71.2	828
21.	many_vox-CLR_coords	73.3	72.6	70.7	73.4	72.5	70.9	73.2	72.7	70.5	831
22.	many_vox-CLR_scale	71.8	71.2	69.6	71.9	71.0	69.8	71.6	71.2	69.2	828
23.	many_vox-CLR_b50	72.3	71.7	70.5	72.4	71.4	70.6	72.1	71.2	70.2	828
24.	many_vox-feature_10	72.2	71.7	69.4	72.3	71.5	69.6	72.0	71.2	69.0	738
25.	many_vox-feature_20	70.9	71.2	68.7	71.0	71.0	69.0	70.7	71.0	68.5	648
26.	many_vox-feature_30	70.9	71.3	69.1	71.0	71.1	69.5	70.7	71.1	69.2	558
27.	many_vox-feature_40	70.3	70.8	68.6	70.4	70.6	68.9	70.1	70.4	68.5	468
28.	many_vox-balance_025	61.4	58.6	57.7	63.3	62.5	58.8	63.2	62.2	58.4	828
29.	many_vox-balance_050	60.2	56.3	56.4	57.4	56.4	52.3	57.3	55.4	52.2	828
30.	many_vox-balance_075	61.8	56.5	58.5	52.7	51.0	48.3	52.8	50.5	48.3	828
31.	many_vox-balance_100	62.2	57.1	58.1	51.0	49.3	46.0	50.9	48.7	46.4	828

Table E.14: Hyperparameter Tuning: Relative Connectivity - Native T1/T2

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	single_vox	63.1	65.1	66.5	63.6	66.0	66.7	63.2	65.5	66.2	92
2.	single_vox-single_targets	63.3	64.2	66.5	63.8	65.3	66.9	63.4	64.7	66.3	1380
3.	single_vox-single_roi	63.5	64.4	59.7	64.0	65.6	59.7	63.7	64.9	59.5	276
4.	single_vox-single_brain	63.5	65.1	66.7	64.0	66.1	67.3	63.7	65.5	66.4	184
5.	single_vox-many_brain	63.2	64.9	66.7	63.7	65.9	67.4	63.4	65.3	66.5	460
6.	many_vox-single_targets	73.9	69.5	69.8	73.5	69.6	69.8	74.1	69.8	69.6	2116
7.	many_vox-single_roi	72.7	69.1	62.2	72.2	69.5	61.9	72.9	69.5	61.9	1012
8.	many_vox-single_brain	70.1	70.5	67.4	69.4	71.1	66.9	70.2	70.8	67.1	920
9.	many_vox-CLR	71.9	71.4	73.0	71.4	71.9	73.1	72.0	71.8	72.9	828
10.	many_vox-CL	70.0	66.1	70.6	69.8	67.7	71.7	70.2	66.2	70.6	828
11.	many_vox-CR	73.1	73.5	71.3	73.1	73.3	70.9	73.1	73.7	71.3	828
12.	many_vox-HLR	68.0	67.4	63.9	68.5	69.3	66.1	67.9	67.8	65.0	828
13.	many_vox-HL	68.8	67.0	63.2	69.6	68.9	67.2	68.9	67.6	65.0	828
14.	many_vox-HR	69.2	68.2	64.3	69.9	70.5	66.3	69.1	68.7	65.0	828
15.	many_vox-HLR-clinical_CAP	69.1	67.9	63.9	69.9	68.1	65.6	69.2	68.0	63.1	829
16.	many_vox-HLR-clinical_UHDRS	68.2	66.9	64.6	68.9	67.0	66.5	68.3	66.9	64.0	832
17.	many_vox-HLR-clinical_all	64.9	62.8	60.7	65.7	62.9	63.2	65.1	63.0	60.7	919
18.	many_vox-CHLR	70.5	70.3	72.1	70.5	71.3	72.7	70.4	70.4	72.0	828
19.	many_vox-CHL	70.8	68.5	70.7	71.1	70.2	72.2	70.8	68.8	70.7	828
20.	many_vox-CHR	71.3	70.8	70.9	71.5	71.8	70.8	71.2	70.9	70.9	828
21.	many_vox-CLR_coords	73.0	71.5	73.0	<b>72.6</b>	<b>72.1</b>	<b>73.3</b>	<b>73.2</b>	<b>71.9</b>	<b>73.0</b>	831
22.	many_vox-CLR_scale	71.2	71.0	72.1	71.0	71.4	72.1	71.3	71.2	72.0	828
23.	many_vox-feature_10	70.3	70.8	71.5	70.1	71.6	71.5	70.4	71.1	71.4	738
24.	many_vox-feature_20	71.5	70.5	71.6	71.3	71.3	71.9	71.6	70.8	71.5	648
25.	many_vox-feature_30	69.4	69.8	70.1	69.1	70.2	70.0	69.6	70.1	70.1	558
26.	many_vox-feature_40	69.9	69.9	70.4	69.8	70.6	70.4	70.1	70.2	70.3	468
28.	many_vox-balance_025	63.1	58.3	63.3	64.6	64.0	64.1	65.1	63.5	63.4	828
29.	many_vox-balance_050	66.5	57.1	63.5	60.6	59.4	56.4	60.8	58.5	55.8	828
30.	many_vox-balance_075	66.8	57.1	64.7	55.8	55.1	52.3	56.1	54.2	51.7	828
31.	many_vox-balance_100	67.6	56.9	64.7	54.7	54.5	50.7	54.9	53.1	50.2	828
32.	many_vox-reinclude	71.5	71.0	71.2	71.3	71.8	71.5	71.7	71.4	71.1	828

Table E.15: Hyperparameter Tuning: Relative Connectivity - Normalized T1

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
1.	<b>single_vox</b>	62.4	64.2	66.5	62.8	65.2	66.6	62.6	64.7	66.2	92
2.	<b>single_vox-single_targets</b>	62.4	64.2	66.5	62.8	65.2	66.6	62.6	64.7	66.2	1380
3.	<b>single_vox-single_roi</b>	63.1	63.9	66.3	63.5	65.0	66.6	63.3	64.3	66.1	276
4.	<b>single_vox-single_brain</b>	63.4	63.0	64.0	63.9	63.7	63.8	63.5	63.2	63.9	184
5.	<b>single_vox-many_brain</b>	63.2	62.4	63.5	63.8	63.2	63.4	63.4	62.4	63.4	460
6.	<b>many_vox-single_targets</b>	71.0	68.5	66.1	70.7	68.9	65.3	71.1	68.6	65.9	2116
7.	<b>many_vox-single_roi</b>	70.5	68.9	69.6	70.4	69.4	69.3	70.6	69.0	69.6	1012
8.	<b>many_vox-single_brain</b>	71.8	69.6	68.0	71.6	69.7	67.3	71.9	69.6	68.0	920
9.	<b>many_vox-CLR</b>	71.6	70.3	71.1	71.7	71.5	70.4	71.7	70.5	70.9	828
10.	<b>many_vox-CL</b>	70.6	68.0	69.9	70.7	70.3	70.0	70.7	68.1	69.9	828
11.	<b>many_vox-CR</b>	72.0	72.1	71.3	72.2	72.0	70.6	72.1	72.3	71.2	828
12.	<b>many_vox-HLR</b>	68.4	63.6	70.0	68.9	65.5	71.8	68.3	63.7	70.3	828
13.	<b>many_vox-HL</b>	66.8	61.9	67.8	67.3	64.4	70.1	66.8	62.1	67.9	828
14.	<b>many_vox-HR</b>	68.9	62.6	69.4	69.7	65.0	71.1	68.8	63.2	69.6	828
15.	<b>many_vox-HLR-clinical_CAP</b>	66.4	66.5	59.3	67.2	66.2	61.7	66.5	66.0	59.1	829
16.	<b>many_vox-HLR-clinical_UHDRS</b>	68.5	66.8	62.1	69.3	66.9	65.0	68.5	66.6	62.2	832
17.	<b>many_vox-HLR-clinical_all</b>	65.4	64.6	62.1	66.2	64.8	64.7	65.4	64.5	62.1	919
18.	<b>many_vox-CHLR</b>	70.0	66.9	71.1	70.2	68.1	71.2	70.0	67.0	71.0	828
19.	<b>many_vox-CHL</b>	69.4	66.1	70.7	69.7	68.0	71.3	69.4	66.1	70.6	828
20.	<b>many_vox-CHR</b>	69.4	67.3	70.2	69.9	68.2	70.2	69.3	67.3	70.2	828
21.	<b>many_vox-CLR_coords</b>	73.8	71.1	71.8	<b>73.6</b>	<b>71.9</b>	<b>70.8</b>	<b>74.0</b>	<b>71.3</b>	<b>71.7</b>	831
22.	<b>many_vox-CLR_scale</b>	71.5	69.5	71.9	71.5	70.6	71.3	71.7	69.7	71.8	828
23.	<b>many_vox-feature_10</b>	70.2	69.8	71.5	70.4	71.2	71.1	70.3	70.0	71.4	738
24.	<b>many_vox-feature_20</b>	71.4	70.3	71.0	71.3	71.3	70.2	71.5	70.7	70.8	648
25.	<b>many_vox-feature_30</b>	72.3	69.9	70.4	72.3	71.1	69.6	72.4	70.2	70.1	558
26.	<b>many_vox-feature_40</b>	70.1	69.8	71.5	70.3	71.0	70.8	70.3	70.0	71.3	468
28.	<b>many_vox-balance_025</b>	67.6	55.9	62.5	67.3	61.7	60.4	67.4	61.0	60.4	828
29.	<b>many_vox-balance_050</b>	63.0	54.4	60.9	58.0	54.9	52.8	58.3	53.8	52.7	828
30.	<b>many_vox-balance_075</b>	61.5	54.9	61.5	50.2	47.8	45.0	50.6	46.9	45.4	828
31.	<b>many_vox-balance_100</b>	64.2	54.9	64.0	49.4	46.4	43.8	49.7	45.2	44.2	828

Table E.16: Hyperparameter Tuning: Relative Connectivity - Normalized T1/T2

ID	Experiment	Raw			Native			Normalized			Input Layer
		Train	Val	Test	Train	Val	Test	Train	Val	Test	
21.	<b>normalized-t1-many_vox-CLR_coords</b>	73.0	71.5	73.0	72.6	72.1	73.3	73.2	71.9	73.0	831
1.	<b>layers_2048</b>	72.7	71.6	72.8	72.4	72.4	73.0	72.9	71.9	72.6	831
2.	<b>layers_1024</b>	71.7	71.7	73.4	71.5	72.6	73.4	71.8	72.0	73.2	831
3.	<b>layers_512</b>	71.9	71.7	72.2	71.6	72.7	72.3	72.0	72.1	72.1	831
4.	<b>activation_sigmoid</b>	71.7	71.7	73.4	71.5	72.6	73.4	71.8	72.0	73.2	831
5.	<b>activation_relu</b>	73.4	71.8	72.1	73.2	72.8	72.3	73.5	72.1	72.0	831
6.	<b>activation_silu</b>	72.5	70.0	70.5	72.2	70.5	70.7	72.6	70.3	70.3	831
7.	<b>activation_elu</b>	71.4	70.9	72.9	71.1	71.6	72.7	71.5	71.3	72.8	831
8.	<b>batch_10e5</b>	69.4	71.1	70.6	69.1	71.4	70.4	69.5	71.4	70.5	831
9.	<b>batch_10e4</b>	71.7	71.7	73.4	71.5	72.6	73.4	71.8	72.0	73.2	831
10.	<b>batch_10e3</b>	73.0	71.7	72.7	72.7	72.5	72.6	73.1	72.1	72.6	831
11.	<b>batch_10e2</b>	70.7	70.5	70.8	70.4	70.8	71.0	70.9	70.7	70.7	831
12.	<b>dropout_000</b>	73.0	71.7	72.7	72.7	72.5	72.6	73.1	72.1	72.6	831
13.	<b>dropout_025</b>	73.5	71.1	73.1	73.2	71.7	73.3	73.6	71.5	73.0	831
14.	<b>dropout_050</b>	72.8	70.6	71.8	72.7	71.4	72.0	72.9	70.9	71.8	831
15.	<b>dropout_075</b>	73.2	71.1	72.7	73.0	72.0	72.6	73.4	71.4	72.6	831
16.	<b>dropout_090</b>	71.2	71.5	71.3	70.9	72.1	71.3	71.3	71.8	71.2	831
17.	<b>patience_4</b>	71.9	71.6	72.4	71.9	72.5	72.5	72.0	71.9	72.2	831
18.	<b>patience_7</b>	73.2	71.1	72.7	73.0	72.0	72.6	73.4	71.4	72.6	831
19.	<b>patience_10</b>	72.8	71.1	71.3	72.6	71.7	71.8	72.9	71.3	71.2	831
20.	<b>rate_10e-4</b>	73.4	71.9	73.4	<b>73.3</b>	<b>72.9</b>	<b>73.4</b>	<b>73.5</b>	<b>72.3</b>	<b>73.4</b>	831
21.	<b>rate_10e-3</b>	73.2	71.1	72.7	73.0	72.0	72.6	73.4	71.4	72.6	831
22.	<b>rate_10e-2</b>	62.4	64.2	66.5	62.8	65.2	66.6	62.6	64.7	66.2	831

Table E.17: Architecture Tuning: Relative Connectivity

# F Source Code

```
source
└── data
    ├── models
    ├── native
    │   ├── preloaded
    │   ├── preprocessed
    │   └── raw
    ├── normalized
    │   ├── preloaded
    │   ├── preprocessed
    │   └── raw
    ├── preprocessed
    └── raw
    └── logs
    └── fsleyes
└── experiments
    ├── subcortical
    ├── diffusion_md
    │   ├── native-t1
    │   ├── native-t1t2
    │   ├── normalized-t1
    │   ├── normalized-t1t2
    │   └── architecture
    ├── diffusion_fa
    │   ├── native-t1
    │   ├── native-t1t2
    │   ├── normalized-t1
    │   ├── normalized-t1t2
    │   └── architecture
    ├── connection
    │   ├── native-t1
    │   ├── native-t1t2
    │   ├── normalized-t1
    │   ├── normalized-t1t2
    │   └── architecture
    └── streamline
    └── misc
```