Using Machine Learning to Fit Whole Brain Models to Concurrent Resting-State EEG and fMRI Data

Andrew Stephen Clappison

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master's degree in Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

ABSTRACT

Whole Brain Modelling (WBM) uses mesoscale computational models of neural activity to study neuroimaging phenomena and medical interventions. Often WBM studies involve selecting a computational model, fitting the model to some neuroimaging data, and then performing experiments on the fitted model. The focus of this thesis is the model fitting aspect, optimizing multimodal WBMs to reproduce phenomena from concurrent resting-state electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) recordings. The goal is to effectively utilize the high temporal resolution of EEG combined with the high spatial resolution of fMRI to generate biologically meaningful parameter sets.

The deep learning methodology used implements WBMs in PyTorch as custom Continuous Time Recurrent Neural Networks (CTRNN), which can be optimized by backpropagation through time (BPTT). This approach has been explored for fitting parameters of the Reduced Wong Wang Excitatory Inhibitory (RWWEI) model to resting-state fMRI (Griffiths & Wang et al., 2022) and fitting parameters of the Jansen-Rit model to evoked potential EEG (Momi et al., 2023). It is different from regular ML approaches, in that each parameter fit has a predefined interpretation that will not change, which adds challenges to finding local optimum regimes, but retains biological interpretability. It has the potential to fit significantly more parameters simultaneously, compared to other approaches such as brute force search. The performance of fitting to resting-state EEG with a power spectral density (PSD) objective function, as well as fitting to true-time scale blood oxygen level dependent (BOLD) signal from fMRI with a novel fitting paradigm is tested. Subsequently, fitting both modalities together in the same objective function is evaluated.

Working with a multimodal model opens the possibility to study the phase relationship between synthetic resting-state EEG and fMRI. This motivates and enables performing a dynamical systems analysis of the alpha-BOLD anticorrelation phenomenon, which relates the phase of empirical EEG and BOLD time series. A case study is presented using the RWWEI model to demonstrate one approach for which models can be used in research once the parameters have been fit. Additionally, the analysis establishes ground truth behavior of the model which can be used to evaluate parameter fitting performance.

ACKNOWLEDGEMENTS

This master's thesis was undertaken as part of a collaboration with the Whole Brain Modelling Group (WBMG), led by my co-supervisor Dr. John Griffiths. The WBMG is part of the Krembil Centre for Neuroinformatics (KCNI) in the Centre for Addiction and Mental Health (CAMH), Toronto. John invests a lot into his student's research and gave me a good deal of responsibility with regards to development of the lab's code base. I would also like to particularly thank the lab member Dr. Zheng Wang, who is the source of multiple ideas my thesis project was built on. Collaborating with the members of the lab was an enriching experience, and attending OHBM 2023 was particularly fun.

At the University of Ottawa, I was part of the Lefebvre Lab, led by my co-supervisor Dr. Jérémie Lefebvre. The research in this lab was focused on the dynamics systems analysis part of my thesis research. Jérémie encouraged me to expand my research focus and provided professional and career advice. I thank the lab members for the interesting discussions regarding their research and journal articles, as well as going to CAN 2022 together.

I am grateful to my co-supervisor Dr. Martin Bouchard for his interest in my project, sponsoring the submission of my thesis to the School of Electrical Engineering and Computer Science (EECS), and coordinating the completion of my degree.

I wish to acknowledge Dr. Maia Fraser for initiating this unique opportunity, which was well aligned with my educational background and research interests, and for providing support. Additionally, I wish to acknowledge Dr. Marcel Turcotte for providing some useful feedback on an earlier version of my thesis.

From my personal life, I would like to mention my appreciation to my parents Stephen and Carolyn as well as to my friend Zihan Wang, for their support in various ways.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ANN: Artificial Neural Network
API: Application Programming Interface
BEI: Balanced Excitation-Inhibition
BOLD: Blood Oxygen Level Dependent
BPTT: Back Propagation Through Time
CAMH: Centre for Addiction and Mental Health
CNMM: Connectome-Based Neural Mass Model
CPU: Central Processing Unit
CTRNN: Continuous Time Recurrent Neural Network
DE: Differential Equation
dMRI: Diffusion Magnetic Resonance Imaging
EEG: Electroencephalography
FC: Functional Connectivity
FFT: Fast Fourier Transform
FIC: Feedback Inhibition Control
FNGFPG: Forward No Gradient - Forward Parallel Gradient
GPU: Graphics Processing Unit
HRF: Hemodynamic Response Function
KCNI: Krembil Centre for Neuroinformatics
LF: Lead Field
LSTM: Long Short-Term Memory
MFM: Mean Field Model
MSE: Mean Squared Error
fMRI: Functional Magnetic Resonance Imaging
ML: Machine Learning
MRI: Magnetic Resonance Imaging
NFM: Neural Field Model
NMM: Neural Mass Model
OOP: Object-Oriented Programming
PSD: Power Spectral Density
RNN: Recurrent Neural Network
RWWEI: Reduced Wong Wang Excitatory Inhibitory
SC: Structural Connectivity
TMS: Transcranial Magnetic Stimulation
TVB: The Virtual Brain
UML: Unified Modeling Language
WBM: Whole Brain Modelling
WBMG: Whole Brain Modelling Group
WhoBPyT: Whole Brain Modelling in PyTorch

LIST OF SYMBOLS

Symbol	Type	Default Value (Units)	Name / Description
S_i^E	State variable	(unitless)	Excitatory population synaptic activity level
S_i^I	State variable	(unitless)	Inhibitory population synaptic activity level
$\boldsymbol{r^E}$	Function Evaluation	(Hz)	Firing rate of excitatory population
$\boldsymbol{r^I}$	Function Evaluation	(Hz)	Firing rate of inhibitory population
$\boldsymbol{I^E}$	Function Evaluation	(nA)	Net input current into excitatory population
$\boldsymbol{I^I}$	Function Evaluation	(nA)	Net input in current into inhibitory population
τ_E	Parameter	$100\ (ms)$	Time constant of excitatory population
τ_I	Parameter	$10\ (ms)$	Time constant of inhibitory population
γ^E	Parameter	$\frac{0.641}{1000}$ (unitless)	Scaling factor of excitatory firing rate, and conversion to per millisecond
γ^I *	Parameter	$\frac{1}{1000}$ (unitless)	Conversion of inhibitory firing rate to per millisecond
σ	Parameter	$0.01\ (nA)$	Magnitude of noise
W_t^E	Noise Variable	(unitless)	Wiener process
W_t^I	Noise Variable	(unitless)	Wiener process
a_E	Parameter	$310\ (nC^{-1})$	Firing rate equation parameter
b_E	Parameter	$125\ (Hz)$	Firing rate equation parameter
d_E	Parameter	$0.16\ (s)$	Firing rate equation parameter
a_I	Parameter	$615\ (nC^{-1})$	Firing rate equation parameter
b_I	Parameter	$177\ (Hz)$	Firing rate equation parameter
d_I	Parameter	$0.087\ (s)$	Firing rate equation parameter
G	Parameter	$1\ (unitless)$	Global coupling strength
J_{NMDA}	Parameter	$0.15\ (nA)$	Synaptic coupling parameter
J_{new} *	Parameter	$1.0\ (nA)$	Synaptic coupling parameter
J_i	Parameter	$1.0\ (nA)$	Synaptic coupling parameter
w_+	Parameter	$1.4\ (unitless)$	Synaptic coupling parameter
I_0	Parameter	$0.382\ (nA)$	Base current into node
W_E	Parameter	$1.0\ (unitless)$	Scaling of I_0 into excitatory population
W_I	Parameter	$0.7\ (unitless)$	Scaling of I_0 into inhibitory population
$I_{external}$	Variable	$0 (nA)$	External stimulus
C_{ij}	Parameter	(unitless)	Connectivity matrix based on subject SC

Sources: Deco et al., 2014; Wong & Wang, 2006
*New symbols introduced

$\boldsymbol{BOLD}$	Function Evaluation		The Blood Oxygen Level Dependent Signal
s_i	State variable		Vasodilatory signal
f_i	State variable		Inflow
v_i	State variable		Blood volume
q_i	State variable		Deoxyhemoglobin content
z_i	Variable		Input signal
V_0	Parameter	0.02	Resting blood volume fraction
k_1	Parameter	2.38	BOLD function parameter based on ρ
k_2	Parameter	2.0	BOLD function parameter
k_3	Parameter	0.48	BOLD function parameter based on ρ
α	Parameter	0.32	Grubb's exponent
ρ	Parameter	0.34	Resting oxygen extraction fraction
τ	Parameter	0.98 (s)	v_i & q_i time constant
κ	Parameter	0.65 (s^{-1})	Signal decay rate
γ	Parameter	0.42 (s^{-1}) **	Flow-dependent elimination rate

Sources: Deco et al., 2013; Friston et al., 2003
** Value of 0.41 used in source
Note: S_i^E is used for z_i.

$\boldsymbol{EEG}$	Function Evaluation	(unitless)	The Electroencephalography Signal
LF_{CxR}	Parameter	(unitless)	Lead field matrix for source space to channel space conversion
N_R	Variable	(unitless)	The vector of neural activity

Note: S_i^E is used for N_i.

CHAPTER 1 – INTRODUCTION

1.1 Motivation

The human brain is a sophisticated system with incredible abilities. As such, it has been a source of inspiration for deep learning algorithms and architectures. At the same time, deep learning techniques can help advance our understanding of the brain itself. One approach for this is using deep learning to improve computational models of the brain, by using advanced techniques to better optimize model parameters. Whole Brain Models (WBMs) represent the neural activity of the brain at the mesoscopic level, using a network of coupled differential equations which define the interaction of excitatory and inhibitory neural populations. Parameter optimization is often an important step when using these models in research.

Successful parameter optimization should result in values for which the WBM replicates multiple characteristics of neuroimaging data simultaneously and accurately, while remaining in a biologically relevant range. To find such parameters, traditional methods such as heuristics, brute force, or other optimization methods can be used. However, deep learning techniques have the potential to fit many more parameters simultaneously than would be otherwise feasible. This comes with the caveat that though more solutions may exist having similar model output, they might not necessarily be biologically relevant.

Given the amount of information being processed in the brain simultaneously, only a small fraction of it is available through various neuroimaging technologies. These technologies have varying degrees of spatial and temporal resolution. So, to help address the overfitting challenge, this thesis works toward the goal of fitting WBMs to multimodal neuroimaging data. In particular, concurrent resting-state electroencephalography (EEG) and functional magnetic resonance imaging (fMRI).

Once a WBM has been fitted to neuroimaging data, it can be useful for various applications. In this thesis, the task is to reproduce and explain resting-state neuroimaging characteristics. That is, phenomena that occur without an external stimulus. A well-known phenomenon that is of particular interest is the alpha-BOLD anticorrelation, which has the potential to relate two very different neuroimaging modalities - EEG and fMRI.

WBMs are also used to understand many physical and mental health problems associated with the brain and their respective treatments. Common applications that these models are used to study are epilepsy (Jirsa et al., 2017), and transcranial magnetic stimulation (Momi et al., 2023). Discoveries from mathematical analysis can help inform biological experiments, thereby saving time, money, and reducing patient risks. Alternatively, these models can be used to identify potential biomarkers of neurological and neuropsychiatric disease processes.

1.2 The Model Fitting Problem

The focus of this thesis is a parameter fitting methodology for Connectome-based Neural Mass Models (CNMMs), such that the simulated output of the model contains the same characteristics and phenomena as empirical neuroimaging recordings. This is not to be confused with a goal of generating the best synthetic neuroimaging data, as perhaps in that case one would discard the CNMM model and choose instead to use a state-of-the-art deep learning architecture. Here, the emphasis is on the CNMM itself, a biologically informed mathematical model of the brain, where the parameters being fit each have some predefined biological interpretation. So, when a CNMM achieves similar phenomena to that found in empirical data it is evidence indicating that the model at these parameter values is valid, while the generated synthetic neuroimaging data itself might not be particularly relevant.

Brute force techniques may be the most straightforward methods to use but are greatly limited by the number of parameters that can be extensively explored. As such, they are often used to fit a small subset of global model parameters. For example, the global coupling (G), global scaling of input into Excitatory population ($w^{E}_{(BG)}$), and the global scaling of input into Inhibitory population ($w^{I}_{(BG)}$) parameters have been fit using brute force techniques (Schirner et al. 2018).

Gradient descent techniques can be used to fit many more parameters. This can be performed either by explicitly calculating the gradients or using more heuristic type approaches to estimate the direction of the gradient (Schirner et al., 2023). Using such techniques, local inhibitory to excitatory coupling values were optimized such that each node achieved a specific mean firing rate for the excitatory population (Deco et al., 2014).

Machine learning techniques such as genetic algorithms (Cakan et al., 2021) or deep neural networks can also be used to fit parameters, with new techniques still being developed. Deep learning uses backpropagation to calculate gradients for gradient descent with respect to some objective function. For example, a deep neural network was used to predict parameters of stochastic differential equations from time series (Gaskin et al., 2023).

In this thesis, we focus on brute force and deep learning-based techniques that take advantage of an understanding of the dynamical systems themselves. The brute force approach uses linear stability analysis to reduce the need to run computationally expensive numerical simulations. The deep learning-based technique is built on work by (Griffiths & Wang et al., 2022). The idea is to backpropagate through the numerical simulation of the CNMM itself, in order to perform gradient descent, as seen in Figure 1.1. Implementing and performing backpropagation through such calculations has recently become more convenient, due to the introduction of PyTorch. This is because if the numerical simulation is implemented in PyTorch, the built-in automatic differentiation functionality can be used to backpropagate through simulation time.

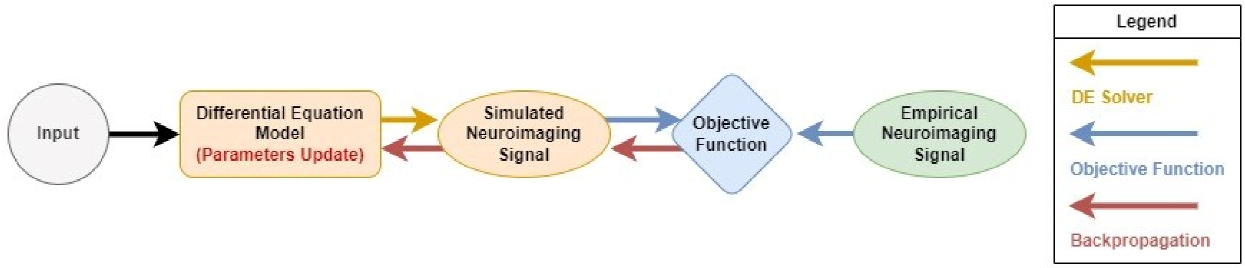

Figure 1.1 – ***CNMM model implementation in PyTorch.*** The model is connected to an objective function so that the error can be backpropagated through time and the gradient used to update model parameters. (Created with draw.io)

Given the nature of the resting-state data being fitted to, for which the only input to the model is noise, the individual time points of the simulated and empirical time series are not expected to match. This is because there is no way to align a phase. However, higher-level characteristics that do not depend on phase like the PSD of EEG (examining the presence of oscillation frequencies) and Functional Connectivity (FC) of fMRI (examining the correlations across brain regions) should be able to match. Thus, various objective function components for fitting these higher order characteristics can come from dynamical systems techniques to analyze differential equation models and techniques to analyze empirical neuroimaging time series themselves.

There are many parameters that can be fit in CNMMs. This includes global or local node parameter values, the network connection strengths, delay times, or parameters of the additional modality modules. Having different local NMM parameters has been shown to improve FC correlation when using gene expression differences related to excitatory and inhibitory ratios to inform local NMM node differences (Deco et al., 2021).

Parameters can be optimized to fit a model to an individual subject's neuroimaging recording, which would be an optimization problem. This could be useful for performing experiments to determine an optimal treatment for an individual patient. Alternatively, parameters can be generalized to work across subjects, which would be a machine learning problem and require a ML paradigm to evaluate the performance. This could be useful for having a model that could predict information about a new patient without a complete neuroimaging dataset. In this thesis, the focus will be on fitting parameters to individual subjects.

The parameter fitting problem is extended in this thesis to the challenge of fitting whole brain models (WBM) to concurrent resting-state EEG and fMRI data. Improving parameter fitting of multimodal CNMMs will progress the field of whole brain modelling. A long-term goal is to have a unified theory of the brain, that captures different time scales and spatial resolutions. It should improve the accuracy of individual model modalities as well. By constraining the model to fit more modalities simultaneously, we hope to identify more biologically relevant parameter ranges. To evaluate the success of the model, decrease in objective function error as well as the ability to reproduce phenomena seen in neuroimaging studies will be measured. For example, the alpha-BOLD anticorrelation, which is a multimodal neuroimaging phenomenon. When model parameters succeed in reproducing neuroimaging phenomena, they can be useful for studying the phenomena themselves.

1.3 Thesis Outline

Chapter 2 provides background on resting-state neuroimaging data, differential equation-based whole brain models, numerical techniques for simulation and analysis, and the open problem of fitting parameters of these models. Parameter fitting is often an important step in WBM research, illustrated in Figure 1.2, for which traditional or deep learning techniques can be used. Fitting can be performed on data from neuroimaging technologies such as EEG and fMRI that capture different aspects of the underlying brain activity. Then, Whole Brain Models can be used to study the underlying mechanisms that generate the phenomena observed in neuroimaging data.

Chapter 3 is a case study of a single node RWWEI model, introducing analytical and numerical methods for working with and interpreting differential equation-based models. First, a technique combining linear stability analysis and grid search is evaluated and used to find parameters sets for which the RWWEI model oscillates. Then, the found parameter sets are used to study the alpha-BOLD anticorrelation question, a cross-modality phenomenon, using dynamical systems analysis.

Chapter 4 is about the implementation of a deep learning technique, used to fit parameters of whole brain models, as a python package. The deep learning technique is reviewed, and the code architecture and *Application Programming Interface* (API) being developed is described. The implementation is based on previous work by the *Whole Brain Modelling Group* (WBMG), with a new RWWEI implementation and some extra features relevant to multimodal parameter fitting. It brings more transparency to techniques and features used in past work, as well as making it more accessible for others to use the deep learning technique in future work.

Chapter 5 is an evaluation of the deep learning technique with new objective functions and parameter fitting paradigms. The goal is to work towards being able to fit a multimodal RWWEI model to concurrent resting-state EEG and fMRI. An objective function is evaluated for fitting source space EEG to a PSD. A new fitting paradigm is evaluated for fitting true-time scale BOLD simulations to empirical FC within computational constraints. Then, some multimodal experiments are performed.

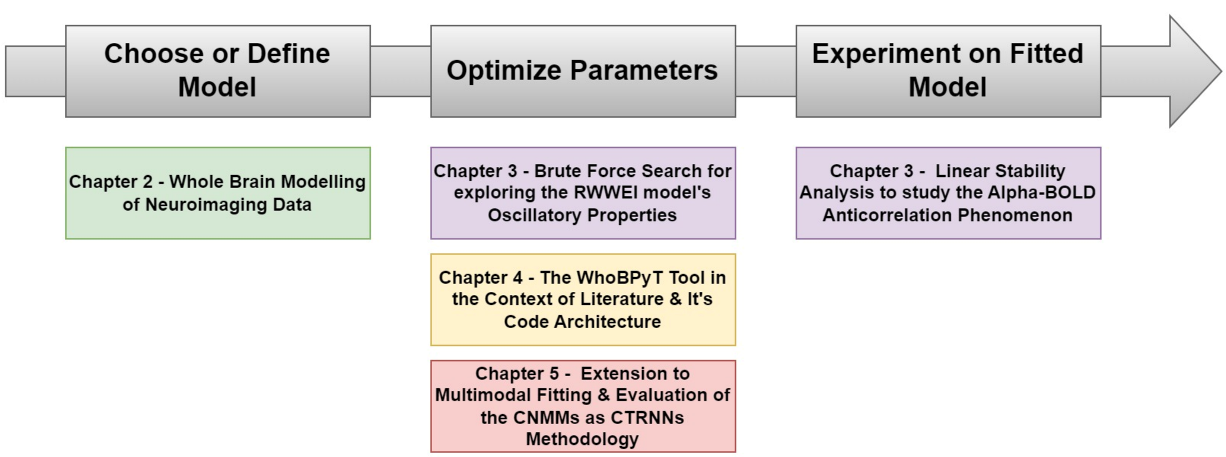

Figure 1.2 – ***Thesis chapters in the context of a computational modelling research pipeline.*** (Created with draw.io)

1.4 Research Contributions

This thesis consolidates the background knowledge required for a deep learning expert to enter the field of WBM. It presents tools for optimizing parameters of WBMs to reproduce resting-state neuroimaging phenomena, as well as methods for interpreting those parameter sets once found. These are applied and evaluated specifically on the RWWEI model.

First, a brute force search using analytical estimations is demonstrated to successfully find parameter sets for which an individual RWWEI node oscillates. Trends in oscillation properties of RWWEI across multi-dimensional parameter spaces are visualized using this technique. Additionally, the BOLD equations were analyzed, to discover that steady state BOLD is largely independent of input frequency and amplitude of a sine wave input. Thus, these two findings lead to a proposed approach for evaluating the alpha-BOLD phenomenon in RWWEI.

The core of this thesis revolves around the creation of an open-source Python package being developed on GitHub by the WBMG. This package generalizes a deep learning technique developed for fitting CNMMs to resting-state fMRI (Griffiths, Wang, et al., 2022) and to *Transcranial Magnetic Stimulation* (TMS)-evoked EEG (Momi et al., 2023). As part of the work for this thesis, a leadership role was assumed to progress the code base from prototypes towards a generalized package. This involved contributions to the architectural design and API of the package, improved documentation in inline comments and in a sphinx website, as well as integrating multiple new features.

This thesis goes beyond previous work by tackling the various challenges of extending the deep learning technique to optimize parameters of a multimodal RWWEI model using concurrent resting-state EEG and fMRI data. Firstly, this required a new idea for fitting to resting-state EEG, so an objective function based on PSD is proposed and evaluated. Additionally, it requires true-time scale BOLD fitting, so a novel fitting paradigm with a reduced computer memory requirement is developed. Finally, some multi-objective problems were evaluated, though further improvements to fitting to individual modalities are required before being able to properly utilize multimodal resting-state data.

In addition to this thesis, the following works have been presented to the following two conferences:

- Clappison AS, Wang Z, Oveisi MP, Momi D, Lefebvre J, Fraser M, Griffiths JD. Whole-brain connectome-based computational modelling of concurrent resting state electrophysiological and hemodynamic activity. CAN2022 (Canadian Association for Neuroscience), May 2022, Toronto. [Poster]
- Clappison AS, Wang Z, Fraser M, Lefebvre J, Griffiths JD. WhoBPyT: A Python Package for Fitting Whole Brain Models to Multimodal Neuroimaging Data. OHBM2023 (Organization for Human Brain Mapping), July 2023, Montreal. [Poster]

CHAPTER 2 – BACKGROUND

2.1 Neuroscience Basics

Cognitive processes are believed to be the result of the interaction of neurons communicating through electrical and chemical signals. Neurons send electrical signals as pulses along their axon, which functions as a wire connecting to other nearby or distant regions in the brain. The electrical pulses are the result of an exchange of ions inside and outside of the cell membrane, and ultimately lead to a release of neurotransmitters at the end of the axon called the synapse, which connects to the subsequent neuron's dendrites (see Figure 2.1). Neurons naturally have a slightly negative voltage between the inside and outside of the cell, called the resting potential, maintained by a sodium-potassium pump. When a neuron receives a sufficient net aggregation of excitatory signal such that it reaches the threshold, an exchange of ions across the membrane results in an electrical pulse, called an action potential, that travels along the axon (Kalat, 2016).

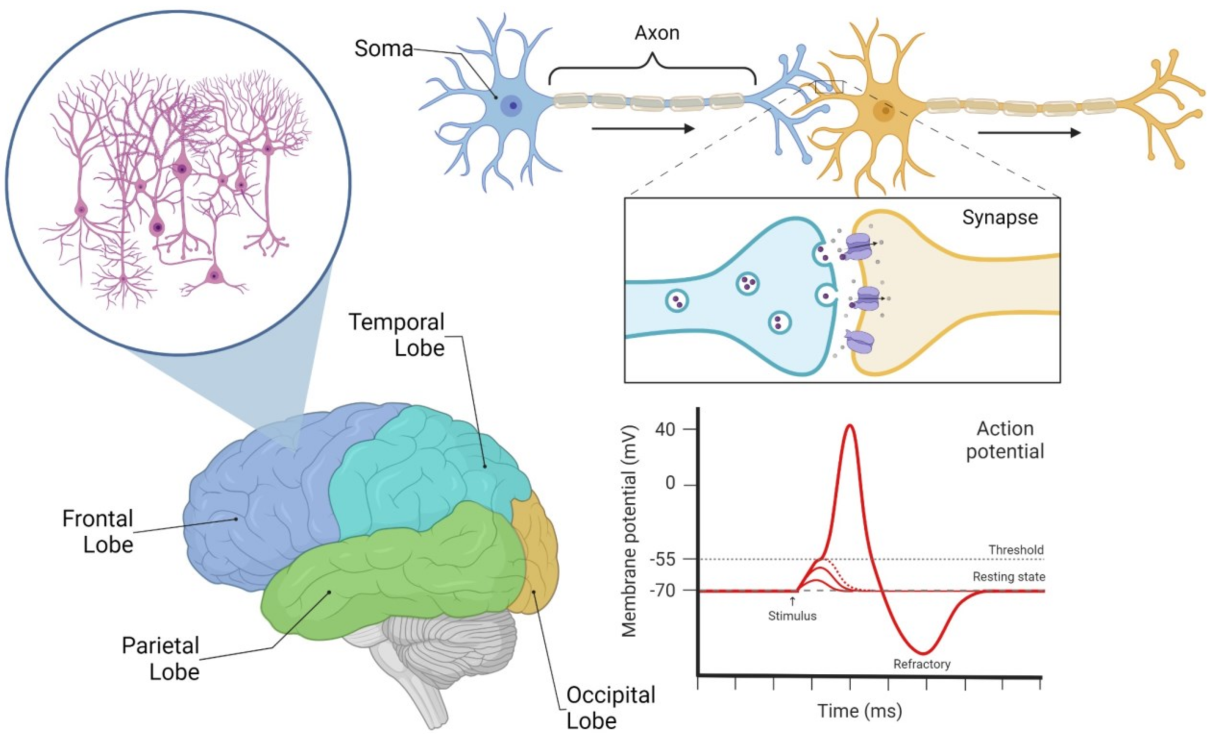

Figure 2.1 – ***Neuroscience background.*** The brain is made up of billions of neurons of various kinds. Neurons interact at synapses, where one neuron's axon connected to another neuron's dendrites. When a neuron receives enough stimulus from other neurons, it causes a rapid increase in potential that travels along its axon. At the end of the axon neurotransmitters will be released which may excite or inhibit the subsequent neurons. (Created with BioRender.com)

Some neuron types send excitatory signals that depolarize the post synaptic cells, while others send inhibitory signals which hyperpolarize the post synaptic cells. This depends on the neurotransmitters released, with NMDA synapses being an example excitatory type and GABA synapses being an example inhibitory type. The brain has adaptive mechanisms which can strengthen or weaken these neural connections to maintain a certain average level of activity and/or for memory and learning.

The human brain is very connected. The cortex can be divided into grey matter and white matter (see Figure 2.2). The grey matter contains many neuron's cell bodies and dendrites. It is about 1 cm thick depending on the location, and it is folded to increase the available surface area. This is where the neural processing is done. The white matter area is made up of axons of neurons, which act as wires sending signals to other areas of the brain. The white part of the white matter is a fatty substance called myelin which surrounds the axons at various places to improve conductivity (Kalat, 2016).

The brain itself is a very structured organ. It has several primary anatomical levels, generally categorized as hindbrain, midbrain, and forebrain. While the hindbrain and midbrain are known to be older from an evolutionary perspective and responsible for more primitive survival functions, the forebrain is known to be responsible for higher level cognition. Different areas of the brain are known to be associated with various functions. In particular, the cortex is divided into frontal, parietal, temporal, and occipital regions. The frontal region is associated with abstract reasoning and contains the motor region. The parietal region contains tactile information. The temporal region is associated with sound. Finally, the occipital region is associated with vision (Kalat, 2016).

2.2 Neuroimaging Data and Resting-State Phenomena

Neuroimaging data comes from technologies which capture the anatomical structure and/or the activity of the brain. There are many different modalities, but common ones are electrical, magnetic, optical, or acoustic based. Here the focus is on concurrent timeseries from electroencephalography and functional magnetic resonance imaging.

While we are primarily interested in resting state phenomena from time series, anatomical (structural) information can support the analysis. For the human brain, this information is often generated from *magnetic resonance imaging* (MRI) and *diffusion magnetic resonance imaging* (dMRI). Standard MRI is able to measure the relative density of different tissues in the brain. From this, a 3D representation of the brain volume can be generated. This volume can be aligned with a brain template to segment it into various regions called parcels. The dMRI measures the diffusion of water, for which diffusion occurs more freely along axon fiber tracks than across them, and so these can be used to generate a connectivity matrix on the strength of connections between each brain parcel (Zhang et al., 2022). These are referred to as *Structural Connectivity* (SC) matrices.

Neuroimaging phenomena are commonly observed abstractions from neuroimaging time series. Typically, a neuroimaging modality such as EEG and fMRI will contain measurements for multiple brain areas at each time point. Common things to look at are the power of various frequencies in the time series, correlations across the different time series, or other abstracted features from the data.

2.2.1 Electroencephalography

Electroencephalography (EEG) measures electrical activity of the brain. In particular, it is believed to measure the electrical activity of the pyramidal neurons which are located in grey matter and have apical dendrites which are parallel to one another and perpendicular to the skull surface (Jackson & Bolger, 2014). This modality is measured by non-invasive electrodes that are placed on the head in various locations. The electrical activity is fast (e.g. short pulses with fast repetition rate), so it has high temporal resolution. However, the signal measured is influenced by multiple areas below the skull, so its spatial resolution is low. EEG measurements at the probe locations are referred to as channel space EEG. The EEG can also be converted into source space, which refers to the predicted spatial locations in the brain where the neural activity being measured by the probes is occurring. This is calculated by using the structure of an individual's head geometry (from MRI) to generate a *Lead Field* (LF) matrix, which can then be used to convert from channel space EEG to source space EEG (Koles, 1998; Montes-Restrepo et al., 2014; Michel & Brunet, 2019). LF matrices are generated based on the structure of an individual or template MRI, which typically accounts for the geometry of the brain and the effect of the skull and soft tissue on EEG recordings.

A common EEG system might measure at 64 sensor locations with a sampling rate of 5000 Hz. Each sensor records one time series. These time series may be bandpass filtered to focus on the frequencies in particular ranges such as 0 - 70 Hz, to remove high frequency noise. Additionally, the time series may be down-sampled as part of preprocessing.

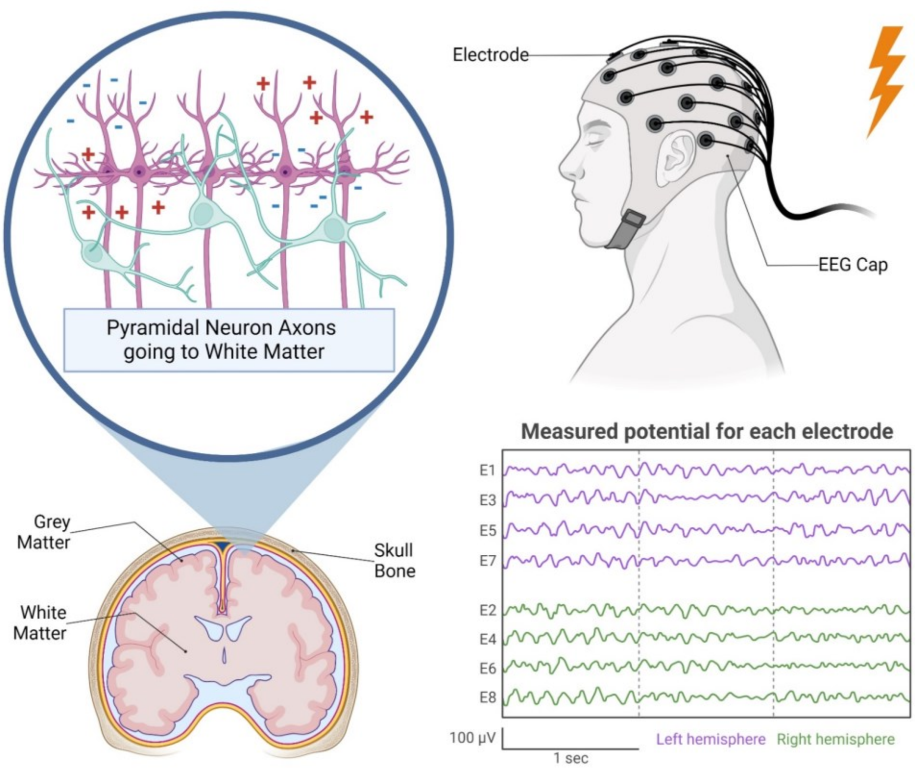

Figure 2.2 – *Modality 1: Electroencephalography.* Pyramidal cells have cell bodies in the grey matter with apical dendrites aligned perpendicular to skull surface. These neurons are believed to be able to generate an electrical signal measured by EEG electrodes. This signal has low spatial resolution, but high temporal resolution. (Created with BioRender.com)

When studying EEG, particularly resting-state EEG, often it is the frequencies present and their associated spectral powers which are of interest. There are various techniques for extracting this information from time series. The *Power Spectral Density* (PSD) is a representation of the power of the various frequencies present in a time series signal. The PSD can be calculated by taking the squared magnitude of the *Fast Fourier Transform* (FFT) of the signal, then normalizing by frequency density. The number of frequency points in the PSD is determined by the length of the data signal, and the range of points in the PSD is determined by the sampling rate of the signal data. Thus, the frequency density depends on both the sampling rate and length of the time series. PSDs can be quite volatile, and so there are various techniques for binning and averaging the PSD of time series. The value at frequency 0 indicates the mean value of the signal, which can be removed from this analysis. Alternatively, one may be interested in how the power changes with time, in which case a time series of PSDs would be generated, which when plotted is called a spectrogram. Other methods can be used to focus on a particular frequency band (envelope of a bandpass filter method), instead of looking at all frequencies simultaneously. This is done by first bandpass filtering the signal to a particular frequency range, then applying the Hilbert transform. This can be done in Python packages like SciPy (scipy.org) and MNE (mne.tools).

One of the first phenomena discovered with EEG was the oscillations at 10 Hz frequency coined as ‘alpha’ by Hans Burger (La Vaque, 1999). The EEG signal can be examined in the frequency domain by performing a FFT and taking the magnitude. Then, alpha is the power associated with the bump around 8-12 Hz in the Power Spectral Density (PSD) (Donoghue et al., 2020). The power spectrum of EEG has since been divided into more frequency bands with various oscillatory ranges. These are delta (1-4), theta (4-8), alpha (8-12), beta (12-30), and gamma (30-70), though the exact boundaries can vary between authors. Interestingly, the power of these different frequency bands in the EEG signal have been associated with various mental states, such as sleep, awake, and alertness (Klimesch, 1999).

EEG alpha power is observed all over the surface of the brain, and this signal can be correlated. It appears strongly in the occipital region when the eyes are closed. It is associated with arousal and the visual system. This alpha peak can vary with age, and is associated with cognitive performance (Klimesch, 1999). Some have shown there are multiple alpha frequencies in the brain, in terms of uncorrelated alpha in the brain and multiple alpha peaks in the same region (Feshchenko et al., 2001).

2.2.2 Functional Magnetic Resonance Imaging

Functional Magnetic Resonance Imaging (fMRI) can be used to measure the *blood oxygen level dependent* (BOLD) signal in the brain. This is a proxy for energy usage, as oxygen is consumed to produce energy. The magnetic signal is measured directly at the brain tissue at high spatial but low temporal resolution. For example, an fMRI image might divide the 3D space of the brain into ~100,000 small cubes called voxels of around 3x3x3 mm each with their own signal but only measured with a frequency of 0.5Hz. The low temporal resolution is less problematic for the BOLD signal as it fluctuates much slower than electrical activity. The spatial resolution of voxels may be too high to be used in analysis so averages of regions of the brain called parcels may be used instead. There are various parcellations used to divide the brain into regions, such as the Desikan-Killiany parcellation (Desikan et al., 2006).

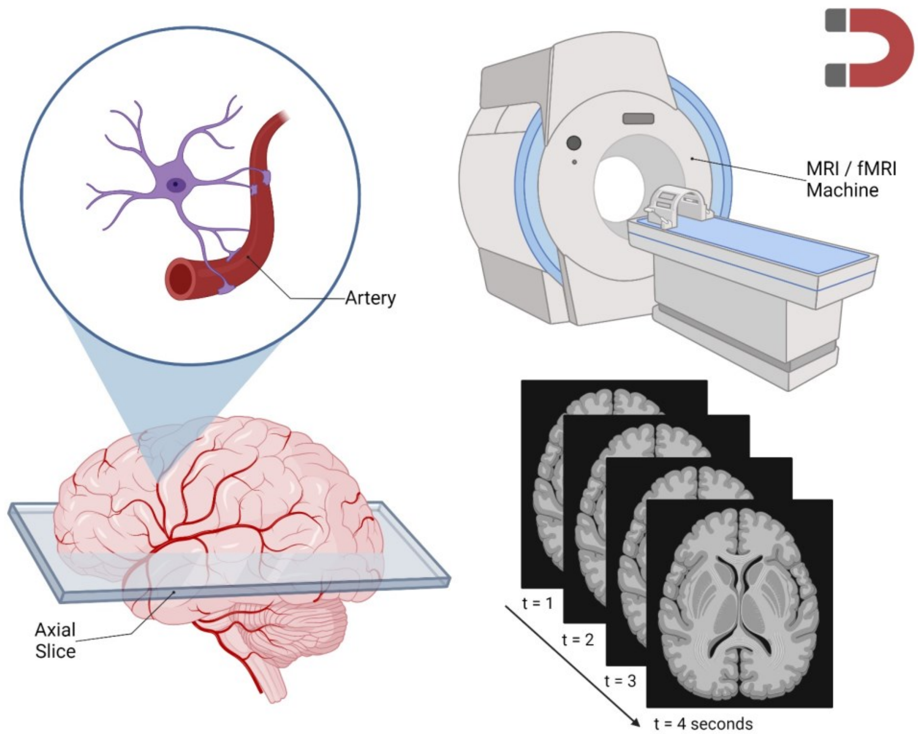

Figure 2.3 – ***Modality 2: Functional Magnetic Resonance Imaging.*** The fMRI Blood Oxygen Level Dependent signal measures changes in oxygen levels that is a proxy for energy use in the brain, as oxygen is consumed to create energy. It has high spatial but low temporal resolution, where only one 3D volume image is generated roughly per second. (Created with BioRender.com)

Often when an fMRI scan is done, an associated MRI image is recorded as well. This MRI image can be used to register the fMRI images more accurately to the same coordinates of some brain template. This is useful to combine with some parcellation file registered to the same template coordinate system. The fMRI signal may need to be parcellated for use in modeling and analysis, so that there is one signal corresponding to each node in the model.

A fundamental phenomenon of fMRI is *Functional Connectivity* (FC). This is the correlation of activity between brain regions, which can be observed during resting-state recordings as well (Biswal et al., 1995). It is related to SC, which measures the physical connections between brain regions (Honey et al., 2007). There are also subnetworks of brain activity seen during resting-state recordings (Damoiseaux et al., 2006).

2.3 Whole Brain Modelling

Whole Brain Modelling (WBM) refers to models that represent the neural activity of interacting brain regions (Griffiths et al., 2021). Neurons are dynamical systems which can be modelled using *differential equations* (DE). Differential equations describe the rate of change of some state variable(s), such as the movement of ions to represent the spiking of neurons. However, in the context of modelling the entire brain, it is too computationally expensive to run a simulation of billions of individual neurons simultaneously. Thus, this motivates abstracting to simplified models which have differential equations that instead represent the activity of populations of neurons.

Models of populations of neurons can generally be categorized into *Neural Mass Models* (NMM), *Mean Field Models* (MFM), and *Neural Field Models* (NFM), though there is some ambiguity in the use of these terms (Moran et al., 2013). NMMs generally have one state variable to represent neural activity, whereas MFMs have a state variable both for the activity and for the variance of the activity and/or other higher order statistics. NFM models on the other hand involve partial derivatives, with respect to both space and time.

We focus on the NMMs, which are simpler representations that can be used for one, multiple, or all regions of the brain. In the case of WBM, representing the entire brain as a network of NMMs, they are called *Connectome-based Neural Mass Models* (CNMM). Each brain region is represented as a node with locally connected excitatory and/or inhibitory neural populations. The connections can have weights and delays based on anatomical properties. Examples of two-state NMMs are the Wilson-Cowan (Wilson & Cowan, 1972) and the Reduced Wong Wang Excitatory Inhibitory model (see Table 2.1; Deco et al., 2014; Wong & Wang, 2006). There are also models with more neural populations in each node such as the Jansen-Rit model (Jansen & Rit, 1995). These models can be connected to other dynamics and equations to represent different neuroimaging modalities. Adding multiple modality modules to an underlying base model can then be done (Kramer et al., 2021; Durstewitz et al., 2022).

CNMM models can be used to produce synthetic data representing various types of activity in the brain, such as electrophysiological or hemodynamic activity. These modalities can have various spatial and or temporal resolutions. The purpose of these models is not simply to produce synthetic neuroimaging data, but rather to generate synthetic data in a biologically interpretable and constrained way. That is, so experiments can be performed on models that have biological significance, and candidate mechanisms for neuroimaging phenomena can be proposed.

There are two main cases where these models are run. That is, task-based experiments and resting-state experiments. In task-based, typically a temporary or period stimulus is added as input to one or more nodes of the model. In this case, the simulated time series may be analyzed directly, as the phase is known. In resting-state, only noise is added to the model which may be in the form of independent additive white noise affecting each population in each node. In this case, time-invariant properties such as correlations between nodes or the signal transformed into frequency space may be examined.

2.3.1 The Reduced Wong Wang Excitatory Inhibitory Neural Mass Model

In this thesis, the *Reduced Wong Wang Excitatory Inhibitory* (RWWEI) two-state model (Deco et al., 2014; Wong & Wang, 2006) was selected for study, which has also been referred to as the *Balanced Excitation-Inhibition* (BEI) model (Aquino et al., 2022). In Deco et al. (2014) the model was applied to study fMRI BOLD, but it has also been used in the context of concurrent EEG and fMRI data (Schirner et al. 2018). It is a simpler NMM model to work with as it only has two-state variables, though there is still more to explore in terms of its oscillatory behavior compared to older two-state variable models like the Wilson-Cowan. Thus, it is used as the primary model of this thesis work.

This RWWEI model was derived from a spiking neural network model with multiple populations (Wang, 2002; Wong & Wang, 2006). The parameter values were chosen based on the parameters from the spiking neurons used to derive the model and then fitting the reaction times of the model to observations in monkey experiments. The model was originally used to represent decision making and working memory of lateral intraparietal cortical neurons. There are some significant differences in the (Deco et al., 2014) compared to (Wong & Wang, 2006). The version by Deco et al., 2014 is what we are referring to as the RWWEI model. The RWWEI version is different as it has an excitatory and inhibitory population instead of two excitatory populations, additive noise is used instead of an Ornstein-Uhlenbeck process, and parameters have different values in general.

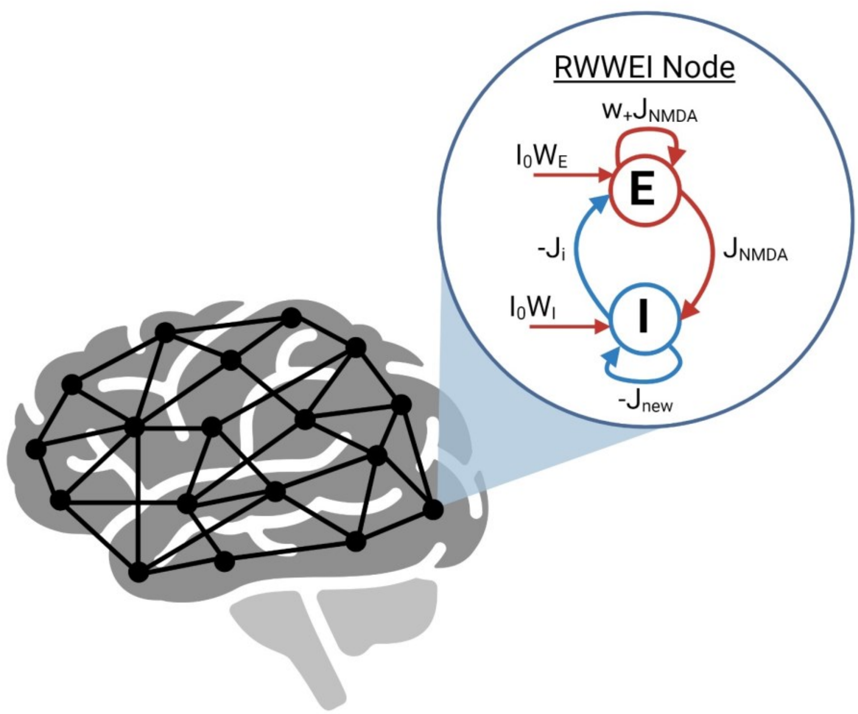

Figure 2.4 – *Connectome-based Neural Mass Models with RWWEI (Deco et al., 2014) Nodes.* CNMMS often contain a network of 10's or 100's of nodes. Each Node has a set of equations with associated parameters, and there are also weights representing the connection strengths between nodes. There may also be extra components to produce different neuroimaging outputs. (Created with BioRender.com)

Connectome-Based Reduced Wong Wang Excitatory Inhibitory Model (Deco et al., 2014; Wong & Wang, 2006)
Each RWWEI node (see Figure 2.4) is defined by two stochastic differential equations representing a population of locally coupled excitatory and inhibitory neuron populations. The excitatory population can be further interpreted as pyramidal cells which are excitatory and contribute directly to EEG recorded signals. For the inhibitory neurons, these can be interpreted as interneurons which are not believed to contribute directly to EEG recorded signals. These neurons are coupled together such that the excitatory population excites itself and the inhibitory population via NMDA receptors, and the inhibitory population inhibits the excitatory population and itself via GABA receptors. The S_i^E and S_i^I variables correspond to the excitatory and inhibitory populations synaptic activity. The activity of the excitatory population is mathematically bounded between 0 to 1, but the inhibitory population is only bounded below by 0 and additive noise can temporarily perturb the state variables outside of these bounds. The model's temporal progression is defined by: $$dS_i^E = \left(-\frac{S_i^E}{\tau_E} + (1 - S_i^E)\gamma^E \boldsymbol{r^E}\left(\boldsymbol{I^E}(S_i^E, S_i^I, I_{external})\right)\right)dt + (\sigma)dW_{i,t}^E$$ $$dS_i^I = \left(-\frac{S_i^I}{\tau_I} + \gamma^I \boldsymbol{r^I}\left(\boldsymbol{I^I}(S_i^E, S_i^I)\right)\right)dt + (\sigma)dW_{i,t}^I$$ Where $dW_{i,t}^E$ and $dW_{i,t}^I$ is the Wiener process (for additive white noise) to node i at time t for the excitatory (E) and inhibitory (I) populations respectively. $\boldsymbol{r^E}$ and $\boldsymbol{r^I}$ are activation functions which convert the net current received by each population into a firing rate: $$\boldsymbol{r^E}(I_i^E) = \frac{a_E I_i^E - b_E}{1 - e^{-d_E(a_E I_i^E - b_E)}}$$ $$\boldsymbol{r^I}(I_i^I) = \frac{a_I I_i^I - b_I}{1 - \mathrm{e}^{-d_I(a_I I_i^I - b_I)}}$$ and $\boldsymbol{I^E}$ and $\boldsymbol{I^I}$ are functions that aggregate input currents received by the excitatory and inhibitory neural populations: $$\boldsymbol{I^E}(S_i^E, S_i^I, I_{external}) = W_E I_0 + w_+ J_{NMDA} S_i^E + G J_{nmda} \sum_j C_{i,j} S_j^E - J_i S_i^I + I_{external}$$ $$\boldsymbol{I^I}(S_i^E, S_i^I) = W_I I_0 + J_{NMDA} S_i^E - J_{new} S_i^I$$ The default parameter values are specified in the Symbol Table. In the single node case, there is no Global Coupling term – highlighted in Grey. That is, $C_{i,j}$ would be the SC matrix in the case of a CNMM, and a 1x1 matrix with value 0 in the single NMM node case.

Table 2.1 – Reduced Wong Wang Excitatory Inhibitory Equations (Deco et al., 2014; Wong & Wang, 2006)

Some RWWEI parameters were biologically informed, and then others were tuned to achieve certain properties during simulation. From the literature, the RWWEI has used a mean firing rate of ~3Hz (Deco et al., 2014) for the excitatory population and a mean firing rate ~8Hz for the inhibitory population (Wong & Wang, 2006), and are within a biologically relevant range for prefrontal cortex in monkeys (Brunel & Wang, 2001; Wilson et al., 1994). To achieve this lower mean firing rate for an unconnected node, the J_{NMDA}, I_0, and W_I were chosen. By changing I_0, and W_I, and noting that W_E is set to 1, this is equivalent to choosing a base input current into the excitatory and inhibitory populations. Subsequently, when added to the network, the *Feedback Inhibition Control* (FIC) method can be used to preserve this low activity. That is, the coupling parameter J_i is tuned individually at each node in the network to achieve a certain mean firing rate of 2.63-3.55 Hz for all local excitatory populations. Thus, similarly, local coupling strengths and base input currents are the parameters being adjusted in this thesis to add additional constraints to the desired model behavior during simulation.

Two parameters have been added to the equations in Table 2.1. The first is J_{new}, which is set to 1 by default. This variable is the local self-inhibition of the inhibitory population, which was not explicitly defined. By adding this parameter, we now have J_{NMDA}, J_i, w_+, and J_{new} which can be adjusted to produce any combination of values for the four local couplings in a single RWWEI node. The second parameter is γ^I, which is set to $\frac{1}{1000}$ and is required as the differential equations are in milliseconds.

Noise is added to both the excitatory and inhibitory populations. In the original RWWEI equations it appears to be written such that both populations of a node get the same noise, but we write explicitly that each node is given independent additive noise. The average magnitude of the noise, however, is constrained to be same for both populations.

2.3.2 The EEG Measurement Model

The simulated EEG signal can either be studied in source space or channel space, as is also the case with empirical EEG data. However, the use of the LF is different from the empirical EEG case, as the model naturally outputs source space EEG, and a parcellated LF matrix can be used to get from simulated source space EEG to simulated channel space EEG (Momi et al., 2023).

Source Space EEG Equation	Channel Space EEG Equation
$\boldsymbol{EEG} = N_i$ N_i: Vector of CNMM Node Activity. Node activity is a function of one or more state variables.	$\boldsymbol{EEG} = LF_{CxR}N_i$ LF_{CxR}: A Lead Field Matrix of Number of EEG Channels by Number of NMM Nodes N_i: Vector of CNMM Node Activity. Node activity is a function of one or more state variables.

Table 2.2 – Source and Channel Space EEG Equations

The chosen representation of the neural activity for N_i may depend on the NMM model, as different models may have different available variables. As an example, the Wilson-Cowan model could use postsynaptic potential, calculated as E – I (Wilson & Cowan, 1972). Here, it is assumed that excitatory neurons are mostly efferent neurons (which pyramidal neurons are), inhibitory neurons are mostly interneurons, and the number of neurons in the excitatory and inhibitory populations are the same (Wilson & Cowan, 1973). In the RWWEI model, there are various options for combining S^E and S^I to represent the source space EEG. One option is to use the strategy from the Wilson-Cowan model, but noting the RWWEI model was derived assuming 160 Excitatory and 40 Inhibitory neurons, so having $0.8 \times S^E$ - $0.2 \times S^I$ as the EEG Signal. Alternatively, one may consider EEG as measuring overall neuronal activity, which would be $0.8 \times S^E$ + $0.2 \times S^I$, again the 0.8 and 0.2 coming from the ratio of Excitatory and Inhibitory neurons the model is based on. However, we will simply use $N_R = S^E$ as a proxy for EEG, because the various options are expected to be highly correlated and using S^E will simplify the methods of this thesis. Noting that EEG is believed to also be related to synchrony, in fact the current proxy that is related to neural activity might be a poor representation. However, most NMM models do not have a variable that captures the synchrony aspect.

Looking at EEG in source space can simplify the problem, as a parcellated LF matrix is not required to project from parcellated source space to EEG channel space. Instead, the S^E state variable can be simply used as a proxy for the electrical activity in that area. This CNMM LF is different from the LF used to transform EEG recordings into fMRI source space, which is instead dealing with many more fMRI voxels instead of NMM nodes. So, in a sense, the LF matrix is parcellated to be compatible with CNMMs in order to produce a synthetic channel space EEG signal.

2.3.3 The BOLD fMRI Measurement Model

For the BOLD signal measured by fMRI, an additional hemodynamic model based on Balloon-Windkessel dynamics is used (Friston et al., 2003). These dynamics are applied to each NMM node individually, as fMRI has high spatial resolution and measures a specific area.

BOLD Hemodynamic Equations (Friston et al., 2003)
The Blood Oxygen Level Dependent (BOLD) signal is defined as follows: $$\boldsymbol{BOLD}(q,v) = V_0\left(k_1(1-q) + k_2\left(1-\frac{q}{v}\right) + k_3(1-v)\right)$$ There are 4 state variables internal to the hemodynamic model. These are s, f, v, and q which represent vasodilatory signal, $$\frac{ds}{dt} = z_i - \kappa s - \gamma(f-1)$$ inflow, $$\frac{df}{dt} = s$$ blood volume, $$\tau\frac{dv}{dt} = f - v^{\frac{1}{\alpha}}$$ and deoxyhemoglobin content, $$\tau\frac{dq}{dt} = \frac{f\left(1-(1-\rho)^{\frac{1}{f}}\right)}{\rho} - \frac{v^{\frac{1}{\alpha}}q}{v}$$ respectively. The equations were applied to a NMM in (Deco et al., 2013), where the synaptic gating variable is used as the input z_i.

Table 2.3 – fMRI Blood Oxygen Level Dependent Signal Equations (Friston et al., 2003)

Note that the time parameters of this model are in seconds, so one must be careful that the dt used by the BOLD equations (with time constants in seconds) is 1000 times smaller compared to the dt of the RWWEI model (with time constants in milliseconds). The simulated fMRI signal has many more time points than empirical fMRI, so down-sampling is often required.

2.4 Numerical Simulations

Differential equation-based CNMM models have now been defined, but the explicit solution for the system is generally unknown. So, to generate synthetic neuroimaging time series from the model, we now transition to numerical methods. We use the Euler method for differential equations and the Euler-Maruyama method (Maruyama, 1955) for stochastic differential equations. While numerically simulated data does not need all the preprocessing steps that empirical data does, there are other factors that need to be accounted for. This includes tuning hyperparameters and choosing appropriate initial conditions. Subsequently, some post processing techniques such as removing the transient period and/or resampling may also be required. The goal of all this is to have good quality synthetic neuroimaging data, for which the same techniques can be applied to analyze empirical neuroimaging data once it has been preprocessed.

The simplest differential equations simulation method is Euler's Method, which works for the non-stochastic case. This method uses a first order approach which estimates the next timepoint based on the current value of system's state variables plus the first derivative multiplied by the step size (which is in units of time for CNMMs). There is an error that accumulates over time which can be reduced by taking smaller steps or using more complicated approaches that integrate information from higher order derivates.

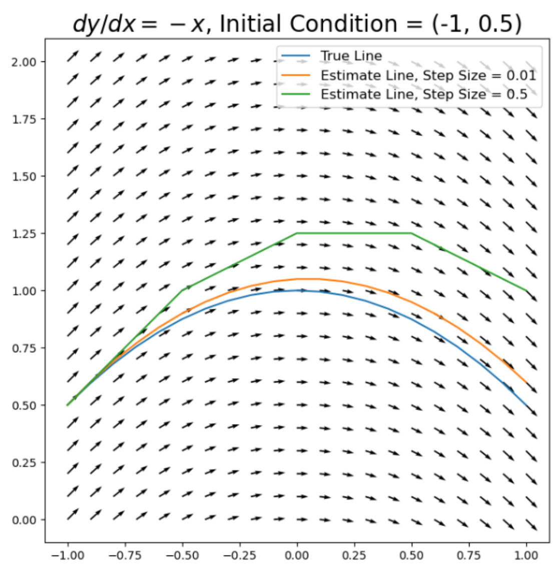

Figure 2.5 – ***Euler simulation of a simple differential equation for a particular initial condition.*** The known solution is drawn in addition to two solutions using Euler's method with different step sizes. Taking smaller steps results in a more accurate representation of the system, but requires more computation. (Created with Python: Matplotlib)

To simulate stochastic differential equations with additive noise, the Euler-Maruyama method can be used. Additive noise is noise that gets added to state variables at each time step independent of value of the state variable itself. This additive noise is of the form of white noise with mean zero and some standard deviation. This method is based on Itô Calculus and can be derived using a Taylor expansion for two variables, where the second variable is the noise (Lee, 2013). That is, because the change in white noise depends on the change in t, but it can be shown that $dW^2 = dt$. Failing to take this into account would result in different input noise amplitudes for different choices of step sizes. For more complex cases of noise, other methods may need to be considered.

Numerical Estimation of Stochastic Differential Equations with Additive White Noise (Euler-Maruyama Method)
We use the Euler-Maruyama Method to evaluate stochastic differential equations with additive white noise. This requires choosing initial conditions, represented as $f(0)$, for the systems state variables. Then, the following formula can be used (for sufficiently small dt) to recursively solve the system across time: $$f(t+dt) \cong f(t) + dt \times f'(t) + \sqrt{dt} \times X$$ $$X \sim Normal(0, \sigma^2)$$ where $f'(t)$ is deterministic part of the stochastic differential equations.

Table 2.4 – Euler-Maruyama Method (Maruyama, 1955)

When running numerical simulations, the accuracy of the simulation can be greatly affected by a few hyperparameters. Possibly the most important hyperparameter is the step size, for which the appropriate choice will depend on the time constants in the equations. The smaller the step size the less the error there will be, but the tradeoff is computational time and possibly also computer memory. For RWWEI, we use a step size of 0.1 milliseconds. Resampling, often in the form of down-sampling, is sometimes required as numerical simulation must be done with high resolution to mitigate error.

Different initial conditions for the state variables can result in completely different model behavior. For example, some initial conditions may lead to the model dampening to a fixed value while others may lead to oscillatory behavior. When examining the oscillations of a particular parameter set, we are interested in the amplitude of the self-sustained oscillation and how robust it is to noise and input from a network of nodes. Thus, for the purposes of determining the power of oscillations in the model, various combinations of state variable values can be considered as initial conditions, and the one with the largest oscillations in amplitude or power are used as the statistic for the model with a particular parameter set. Alternatively, to save computation time, the fixed points (see Section 3.2.1) can be determined and used as the initial conditions (with a small initial perturbation in the non-stochastic case), as will be described in the next chapter.

Transient periods are also a result of the choice of initial conditions. There can be a transient period for which the state variables trend towards particular values and then stay there, or it can dampen with oscillation towards particular values and stay there without noise. This is problematic if not removed, as it can cause false readings when subsequently calculating oscillation power or amplitudes. Thus, it is often useful to discard some initial portion of the simulation.

An example simulation of RWWEI using Euler-Maruyama Method is shown in Figure 2.6.

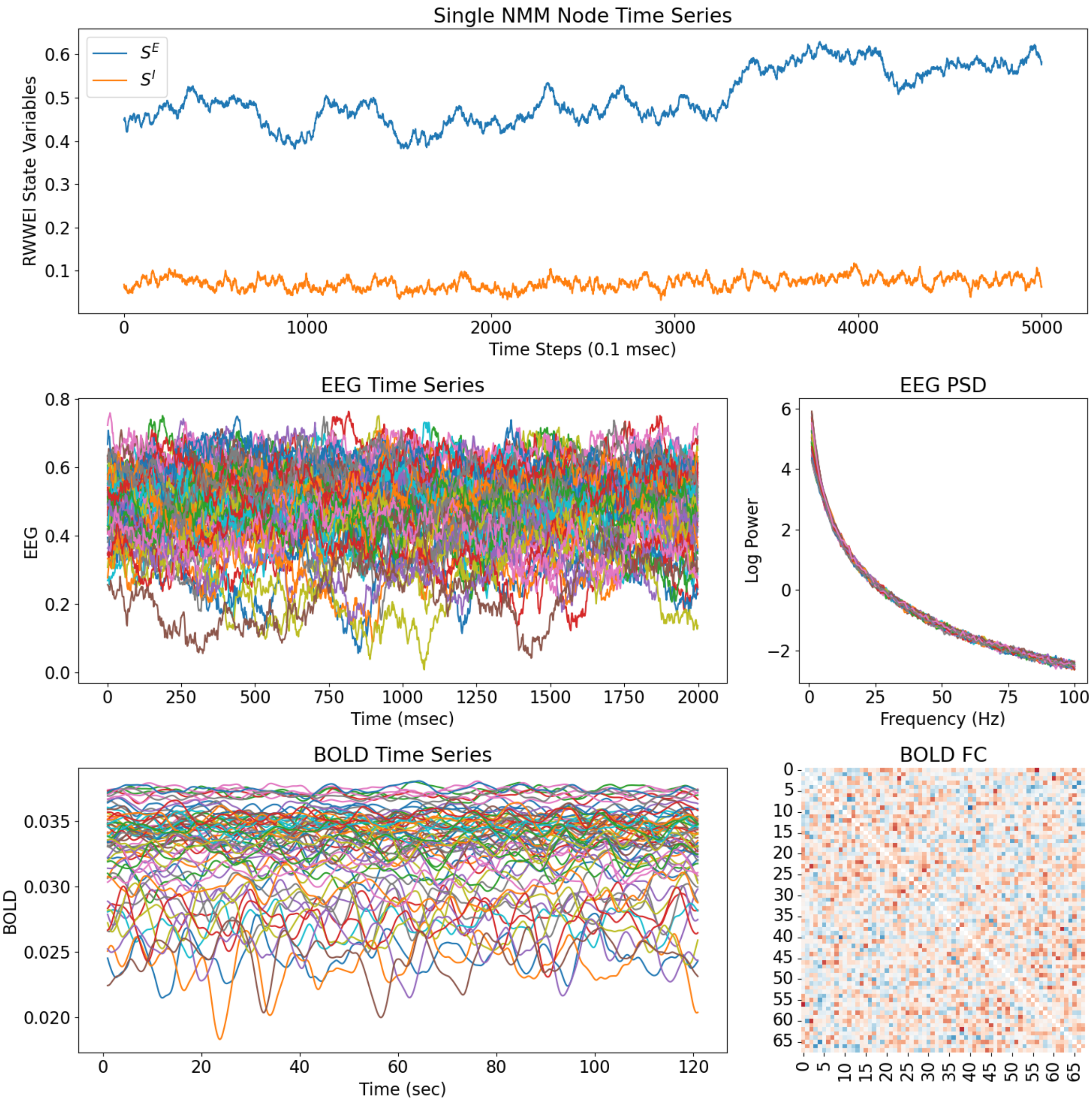

Figure 2.6 – *Full Multimodal RWWEI Numerical Simulation.* Simulation of RWWEI using default parameters except G = 0.5. The SC used was the average SC of the Deligianni and Clayden Dataset (Deligianni et al., 2016), normalized by the matrix 2-norm. Five minutes of simulation was run with the first 30 seconds removed as transient. **Top:** Shows a segment of time for the state variable values of one RWWEI node in the network. **Middle:** Shows a segment of source space EEG from the model simulation with an associated PSD. There are no dominant frequencies observed using this parameter set, which would appear as bumps in the PSD (see Figure 5.6). **Bottom:** Shows a segment of BOLD time series and an associated FC matrix (red is positive correlation, blue is negative correlation). A good quality FC structure would have strong positive correlations in the top left and bottom right of the matrix representing the two hemispheres, as well as diagonal components offset from the middle diagonal representing cross hemisphere communication. Here, a correlation structure is only weakly observed. (Created with Python: Matplotlib, Seaborn)

2.5 Existing Software for Whole Brain Modelling

There are multiple software packages available for running CNMM model simulations, some which also include parameter fitting functionality. This includes the well-known *The Virtual Brain* (TVB), as well as more niche packages like Neurolib. There are other software options where the CNMM model equations need to be implemented as well, but for which numerical integration is taken care of. Alternatively, it may be valuable to create one's own implementation, as small implementation variations could have large impacts on the results.

The TVB package has built-in implementations of many different CNMMs (Sanz Leon et al., 2013). The output of such models is synthetic neuroimaging data. It also provides functionality for analyzing the model with analytical tools. As a consequence of adding more biological details, preparing data for input into TVB is a more involved task due to the multiple input files expected to be used for various features.

Neurolib is another example of a python library that provides CNMM implementations (Cakan et al., 2021). It has a simpler interface and provides functionality for using a genetic algorithm to fit model parameters. For simulations, only connection strengths and distances (which can be set to zero if not using delays) are required to run a CNMM simulation.

There are libraries and software packages with general differential equation solvers for which CNMMs can be manually implemented in. Additionally, there are some ODE solvers implemented in Python such that if the model is implemented parameter fitting methods using an adjoint method can be used. This includes torchdiffeq (Chen et al., 2018), torchsde (Li et al., 2020), and torchdyn (Poli et al., 2021). Alternatively, a statistical approach would be automatic differentiation variational inference in STAN (Kucukelbir et al., 2017), which can estimate the gradient using statistical techniques.

WhoBPyT can be viewed as a complementary tool to these other existing software packages, proving an implementation of a parameter fitting technique, for which the found parameters can then be used in verification NumPy implementations or other software afterwards for experimentation.

2.6 Recurrent Neural Networks

Artificial Neural Networks (ANN) are data processing architectures inspired by neurons in the brain. Broadly speaking, ANNs can be categorized into those with only feed forward connections and those which have recurrent connections. *Recurrent Neural Networks* (RNN) are a common architecture used for working with or generating sequential data, such as text or time series. This is related to the fact that having a recurrent component in a model means that it can retain some memory state, which depends on the initial state and subsequent inputs.

When backpropagation was applied to feed forward ANNs in the 1980's (Rumelhart et al., 1986) for updating network weights, these architectures became more widely studied and useful for practical applications. Backpropagation was quickly applied to various model architectures including RNNs as well (Pineda 1987; Almeida 1987).

Some variations of ANNs are known to be universal function approximators (Hornik et al., 1989), and of particular interest here is that RNNs can approximate dynamical systems (Funahashi & Nakamura, 1993). In fact, RNNs have a close relationship to dynamical systems, where they can be derived from the discretization of differential equations (Sherstinsky, 2018). Example continuous and discrete RNNs are shown in Table 2.5.

RNN Type	Parameters	Comments
Continuous Time RNN (Pineda, 1987) $\frac{dx}{dt} = -\alpha x + \beta f(Wx) + I$ $f(y) = (1 + e^{-y})^{-1}$	α : 1 β : 1 x : vector of neural activities f : a function such as a logistic function W : connection strengths I : vector of external input bias	Example continuous time RNN which is defined using differential equations.
Elman RNN (Elman, 1990) $h_t = \tanh(Wx_t + Uh_{t-1} + b)$ $output_t = f(h_t)$	x_t : the input at time t h_t : is the hidden state at time t W : weights to multiply input U : weights multiply hidden state b : bias term f: any function of choice	Example discrete time RNN which has a fixed step size.

Table 2.5 – Example Variations of Continuous and Discrete Recurrent Neural Networks

A special case of RNNs is when the forward steps represent time steps. In this case each pass through the RNN is a step forward in time by discrete time steps. Thus, backpropagation in this case is *backpropagation through time* (BPTT) (Werbos, 1990). This makes RNNs naturally suited for neuroimaging data, which is often in the form of time series. For high dimensional data, the input will typically need to be encoded before being fed into an RNN. For example, when working with fMRI time series, which is a time series of 3D images, it can be reduced to a time series of average signals from different parcellated brain regions. In this case, there is a time series of vectors where each element in the vector represents the average activity of one brain parcel.

RNNs are known to have a problem training on longer time series, due to vanishing or exploding gradients. There are variations of the RNN such as *Long Short-Term Memory* (LSTM) created to solve this problem (Hochreiter & Schmidhuber, 1997). Alternatively, there are solutions such as truncated back propagation through time to address this as well (Williams & Peng, 1990), where only a fixed number of recent time steps are backpropagated through.

A separate but related problem with backpropagation is that it can require a lot of memory for longer input sequences. There are alternatives to backpropagation for RNNs, such as real time recurrent learning, also referred to as forward propagation (Pearlmutter, 1995; Williams & Zipser, 1989; McBride & Narendra, 1965). However, this technique is more computationally intensive than backpropagation for large networks.

New RNN variations are continuing to be developed. For example, Liquid Time Constant RNNs (Hasani et al., 2021) employ a differential equation representation of the RNN that has variable time constant, depending on the state of the network. They have been trained for an autonomous driving task, by backpropagating through a semi-implicit ODE solver (Lechner et al., 2020).

Another recent study evaluated Harmonic Oscillator RNNs inspired by oscillations in the brain. This network is similar to a CNMM and was used to classify MNIST digits (Effenberger et al., 2022). It showed better performance when the nodes of the network were non-homogeneous, providing additional motivation for fitting local node parameters in the present work, which is explored in Section 4.3.

Standard RNNs can be used to generate synthetic neuroimaging time series directly, though this would not have a clear biological interpretation for the trained weights. Regardless, they can still be used to study the brain in terms of analyzing parameters and temporal and geometric properties (Durstewitz et al., 2022). This leads to the next section, which not only increases the interpretability of the parameters, but also uses the CNMM equations to support their optimization.

CHAPTER 3 – CASE STUDY: ALPHA-BOLD ANTICORRELATION IN A SINGLE NMM NODE

Abstract

This case study examines the alpha-BOLD anticorrelation phenomenon, a cross-modality neuroimaging phenomenon relating the alpha power from EEG to the BOLD signal from fMRI. This is a challenging but interesting topic to study with multimodal NMMs, as it crosses both modalities and time scales. The analysis is performed using a single node Reduced Wong Wang Excitatory Inhibitory (RWWEI) two-state model (Deco et al., 2014). This model has previously been used for concurrent EEG and fMRI research (Schirner et al., 2018). The properties of the model at the individual node level will also be examined more broadly, though in subsequent chapters the focus will instead be looking at networks of nodes informed by brain anatomy.

The first task is to use and evaluate an analytical technique for finding appropriate parameters such that the individual RWWEI node oscillates within a desired frequency band. That is, a technique that estimates oscillation frequency using linear stability analysis in a grid search. This technique is found to work well, but alpha oscillations seem to be sensitive to input, such that small negative inputs can eliminate oscillations entirely and small positive changes in input can lead to more robust beta or gamma frequencies.

Subsequently, the BOLD dynamics are analyzed directly, to examine its contribution to an alpha-BOLD anticorrelation. The steady state responses to sine wave inputs varying in frequency, amplitude, and means were examined. It was found that the BOLD signal was dependent on mean input, but changes in frequency and amplitude were relatively insignificant. This justified a simplifying assumption, to reduce the alpha-BOLD question to looking for an alpha power vs. steady state synaptic activity anticorrelation in the model.

Finally, a dynamical systems analysis using the found parameters is performed, looking for underlying structures by examining the fixed points for varying input currents. Several candidate alpha-BOLD mechanisms are discussed, but due to various limitations that are described, they are only hypotheses.

In going through this case study, challenges of brute force optimization approaches are highlighted, providing motivation for new optimization and ML techniques presented in later chapters. That said, traditional methods still have a role to play, as deep learning techniques do not replace the need to have a good understanding of the model itself. These techniques can be useful for determining appropriate initial conditions of model parameters for training, and of state variables for simulation. However, more importantly these techniques are used for explaining the behavior of NMM models once relevant parameters have been identified.

3.1 Introduction

This chapter is a case study to demonstrate how NMMs, in particular multimodal NMMs, can be used in neuroscience research with analytical methods. The investigation will be focused on the alpha-BOLD anticorrelation, a commonly observed phenomenon in concurrent EEG-fMRI neuroimaging recordings. One particular NMM, the RWWEI model (Table 2.1; Deco et al., 2014; Wong & Wang, 2006) will be used for discovering candidate mechanisms of the phenomenon. This is a natural question to study with multimodal NMMs, as they allow for the representation of multiple modalities and time scales.

3.1.1 Alpha-BOLD Anticorrelation

The alpha-BOLD anticorrelation is a phenomenon seen in concurrent resting-state EEG and fMRI experiments. Alpha is the near 10Hz electrical signal observed in EEG recordings, for which the power fluctuates over time. The BOLD signal on the other hand is a proxy for metabolic activity in the brain. The phenomenon states that an increase in alpha power is associated with a decrease in BOLD activity. This is brain region specific, with most correlations being negative in frontal, parietal, occipital, but in fact some brain regions (thalamus and sometimes insula) show a positive correlation instead (Goldman et al., 2002; Laufs et al., 2003; Moosmann et al., 2003).

Competing theories exist that alpha oscillations are either generated by the cortex itself (particularly the occipital cortex) or originate from the thalamus (Hutt & Lefebvre, 2020). In fact, unlike many other regions of the brain, the thalamus has a positive correlation with alpha power (Goldman et al., 2022). There are other studies which measure the effect of having an input with alpha frequency. It has been shown in an EEG and fMRI study of human participants that having a visual stimulus oscillating at an individual's intrinsic alpha frequency was correlated with certain subnetworks in the individual's FC (Jaeger et al., 2023). There has also been a study having a 10Hz, 40Hz, and 100Hz stimulus in the thalamus of rats, and found that the 10Hz stimulus decreased cortical activity in areas whereas the higher frequencies increase it (Liu et al., 2015). However, another study injected current directly into the thalamus of an anesthetized mouse brain and found that a frequency of 20Hz resulted in highest BOLD response and either increasing or decreasing the frequency trended to lower BOLD response (Lai et al., 2015). Noting, however, that this may be confounded by the anesthesia state.

There are various ways that this phenomenon is measured empirically. In concurrent EEG and fMRI datasets, often the EEG time series are converted to time series of the power of alpha and convolved with a hemodynamic response function before correlating with BOLD (Pang & Robinson, 2018). The convolution with the hemodynamic response function may have a similar effect as to adding a delay. Then, the time series needs to be down-sampled to a frequency which matches that of the fMRI. The alpha power and BOLD are naturally fluctuating over time during resting-state recordings, and so it is a statistical anticorrelation that is observed. The result may be an EEG channel by fMRI voxel matrix of correlations which is quite large, where each row can be presented as a 3D plot for an EEG channel.

The alpha-BOLD anticorrelation is counter-intuitive, as one may think that increased power of oscillations would be associated with higher energy use, thus a higher BOLD value. However, it could be instead that EEG is measuring synchrony, and that perhaps it takes less energy for the neurons to spike in synchrony than to spike irregularly. Next, we examine some published computational studies that have attempted to model and provide an explanation for this phenomenon.

3.1.2 Prior Modelling Studies

Computational models have already been used for studying the alpha power vs. BOLD anticorrelation. Multimodal models are particularly useful for this question because if they can be put into a state where alpha-BOLD anticorrelations are observed, then various technique can be used to explain the presence of the alpha-BOLD anticorrelation in the model. These explanations can then help direct further wet lab studies.

One study used a NMM to examine the effect of injected alpha oscillating current with fluctuating power into a numerical simulation. The NMM used was the Reduced Wong Wang Excitatory Inhibitory (RWWEI) two-state model (Deco et al., 2014), BOLD generated using the Balloon-Windkessel model, and EEG was interpreted to be in source space. The model was individually fit to 15 different subjects of data, where three global parameters of G, ω_{BG}^{E} and ω_{BG}^{I} were fit using a brute force method, with local J_i's fit using a FIC method where EEG in source space was the input into the model instead of noise (Schirner et al., 2018). Subsequently, an experiment was done that injected alpha with varying power into the model, and this resulted in the alpha-BOLD anticorrelation phenomenon being observed. The resulting hypothesis is that the alpha-BOLD anticorrelation is a result of the upper part of the alpha wave causing higher firing rates with increased power, but the lower part being bounded by a firing rate of 0 when at the lower part of the alpha wave, thus resulting in a net increase in the inhibitory population which can dominate and inhibit the BOLD signal through the excitatory population.

Another study proposed that the alpha-BOLD anticorrelation is due to the fact that alpha oscillations are at a higher frequency than the BOLD signal, so when looking at a power spectrum if one goes up it will be correlated with the other going down (Pang & Robinson, 2018). This study used the Robinson model, which is a cortico-thalamic neural field model (NFM). The model was fit to EEG and fMRI, and then examined to fit corticothalamic loop gains and power spectrum. The data used was an EEG dataset, where the frequency component from 0-4Hz was used as a proxy for BOLD.

A critique of these approaches is that their arguments seem to apply to any higher frequency oscillations that are not necessarily required to be specific to the alpha band. Furthermore, using low frequency EEG as a proxy for BOLD may introduce some circular reasoning. Perhaps the anticorrelation mechanism is unrelated to the neural mechanisms that produce the alpha peak in the EEG power spectrum, and in fact applies to lower frequencies as well. However, it is well reported that there is something unique about the presence of alpha peak in the power spectrum of EEG, so it seems reasonable to hypothesize that these two phenomena would be related. Regardless, the question of why an alpha peak exists and why alpha-BOLD anticorrelation exists are still open questions to be further studied.

The alpha-BOLD phenomenon may be a local phenomenon or a result of global connectivity, among other possibilities. NMMs are typically used in the context of connectome-based whole brain modelling where multiple nodes are connected. Additionally, for one population of neurons it may be more appropriate to use a higher resolution model where each neuron is individually modelled and connected with other neurons. Regardless, for the purposes of this case study a single NMM node will be analyzed. In particular, to see if the alpha-BOLD phenomena can be found in an NMM by adjusting the local coupling strengths, and if so, use the model to determine a candidate mechanism.

3.1.3 This Case Study's Approach

In this alpha-BOLD case study, a single RWWEI node (Table 2.1) with parameter regimes that produce innate alpha oscillations is examined. These parameter regimes can be found by varying local coupling strengths and base input values. However, before narrowing in on the alpha-BOLD question, a method for finding parameters for which the model oscillates at various frequencies is examined, with the associated frequencies the model can produce in general. Sensitivity analyses, both analytical and numeric, are performed to see how robust the oscillations are to changes in $I_{external}$ in terms of affecting the frequency, mean, and amplitude of oscillations and the corresponding BOLD signal. Finally, narrowing in on the alpha-BOLD question, a parameter set with alpha oscillations is analyzed to look for candidate mechanism(s) that can explain the alpha-BOLD phenomenon. As this is a case study to demonstrate various techniques, analysis is performed using analytical and numeric techniques for both the stochastic and deterministic cases.

Parameters can be found that produce self-sustained alpha oscillations in an isolated RWWEI node. In the past, oscillations from the Wilson-Cowan model have been studied by adjusting the time constants of the excitatory and inhibitory populations (David & Friston, 2003), but here we study changes in the local coupling strengths and base inputs into the excitatory and inhibitory populations, as well as the base input current into each population. We focus on six parameters of interest (J_{NMDA}, J_i, J_{new}, w_+, W_E, W_I). This is done using a clever brute force approach, that uses analytical methods to filter to parameter sets with a high probability of having alpha oscillations. This reduces the need to run computationally expensive simulations (for various initial conditions) so that wider parameter searches can be performed.

The search here focuses on the deterministic case which is simpler to work with. This approach is viable due to two observations of the model. Firstly, if the model is innately oscillating with constant input, then it is generally doing so in a periodic way. Secondly, that steady state BOLD depends essentially on the mean value of the oscillation, which can be considered constant for periodic oscillation and thus corresponds to a steady state BOLD. Thus, different values of $I_{external}$ can be used for which alpha power and BOLD steady state can be calculated. In this case, the alpha-BOLD anticorrelation is searched for by looking for an inverse trend or negative correlation over a range of $I_{external}$ of approximately 1 nA.

This approach uses the same NMM model as in the (Schirner et al., 2018) study, but instead of injecting oscillating current into the model, parameter regimes are found for which the model naturally oscillates at alpha frequency. Then, the model BOLD is perturbed to different steady state values. So, in a sense this is the opposite of what was done in Schirner et al., 2018. As we are looking at steady state values, we do not need to use a *Hemodynamic Response Function* (HRF) on EEG alpha power before comparing to BOLD, as the time lag the HRF would apply is not relevant for pairs of steady state activity.

First, we address the single modality question of the model of being able to demonstrate the alpha phenomenon. That is, an assumption is made that the alpha-BOLD phenomena must occur in a parameter space where alpha oscillations are present. There are multiple ways an alpha-BOLD anticorrelation could manifest in empirical data. Here we are looking for a smooth trend in alpha power vs. BOLD over a range of 1nA. The ideal hypothesized mechanism would be unique to alpha. In the next section, we review the concept of linear stability analysis at fixed points which is relevant both for the parameter search technique as well as for the analysis of the equations in the found parameter regimes.

3.2 Methods

In this case study, linear stability analysis techniques are focused on, with some numerical analysis performed to confirm the accuracy of the techniques for a single node RWWEI model. Numerical methods have already been introduced in Chapter 2, so the focus here will be on the analytical techniques. This includes techniques to predict oscillating behaviour, to find parameter sets that result in oscillations, and to measure the power and change in oscillations.

3.2.1 Identifying Fixed Points

Fixed points are combinations of a model's state variables, such that without external perturbation the state of the system will remain unchanged. In the context of NMMs, this is saying that the firing rates of the excitatory and inhibitory populations will remain constant with respect to time. Particular to RWWEI, fixed points are the values of S^E and S^I satisfying the implicit equations:

$$\frac{dS^E}{dt} = 0 = -\frac{S_0^E}{\tau_E} + (1 - S_0^E)\gamma^E \boldsymbol{r^E}\left(\boldsymbol{I^E}(S_0^E, S_0^I, I_{external})\right) \tag{2.1}$$

$$\frac{dS^I}{dt} = \ 0 = -\frac{S_0^I}{\tau_I} + \gamma^I \boldsymbol{r^I}(\boldsymbol{I^I}(S_0^E, S_0^I)) \tag{2.2}$$

Identifying fixed points is not straightforward as we do not have an explicit solution for identifying them. Furthermore, depending on the model parameters there may be a different number of fixed point. Thus, a brute force approach is used which exploits the fact that S^E and S^I are only biologically relevant in the range of 0 to 1 which represents 0% to 100% synaptic activity (though additive noise can temporarily push the state variables outside of this range during simulation). This is done by uniformly sampling from the 2D space of S^E in [-1.1, 1.1] and S^I in [-1.1, 1.1] and using these points as initial conditions in a solver such as the fsolve() function from scipy.optimize. The solver will attempt to find a fixed point near each initial condition, and then the solutions are rounded and duplicates are removed. The sampling of S^E and S^I must be chosen with high enough density to insure all fixed points are identified. Additionally, it is necessary to re-test the fixed points in the RWWEI equations directly, as numerical stability issues can occur during these calculations.

3.2.2 Linear Stability Analysis

Next, we can learn about the dynamics near a fixed point by calculating and examining the eigenvalues of the Jacobian matrix (the matrix of partial derivates) of the system at the fixed point. The Jacobian matrix is a linear approximation of the system examined at a fixed point. After computing the Jacobian for particular model parameters at a corresponding fixed point in Python, the eigenvalues of the system are calculated using the eig() function from numpy.linalg. The Jacobian $\boldsymbol{A}$ of the RWWEI system (Table 2.1) is calculated as follows:

$$\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2.3}$$

For which each A_{mn} are as follows:

$$A_{11} = \frac{\partial}{\partial S^E}\frac{dS^E}{dt} = -\frac{1}{\tau_E} - \gamma_E \boldsymbol{r^E}\left(\boldsymbol{I^E}(S^E, S^I, I_{external})\right) + (1 - S^E)\gamma_E \frac{\partial r^E}{\partial S^E} \tag{2.4}$$

$$A_{12} = \frac{\partial}{\partial S^I}\frac{dS^E}{dt} = (1 - S^E)\gamma_E \frac{\partial r^E}{\partial S^I} \tag{2.5}$$

$$A_{21} = \frac{\partial}{\partial S^E}\frac{dS^I}{dt} = \gamma_I \frac{\partial r^I}{\partial S^E} \tag{2.6}$$

$$A_{22} = \frac{\partial}{\partial S^I}\frac{dS^I}{dt} = -\frac{1}{\tau_I} + \gamma_I \frac{\partial r^I}{\partial S^I} \tag{2.7}$$

with:

$$\frac{\partial r^E}{\partial S^E} = \frac{\left(a_E \frac{\partial I^E}{\partial S^E}\left(1 - e^{-d_E(a_E I^E - b_E)}\right)\right) - \left((a_E I^E - b_E)\left(-e^{-d_E(a_E I^E - b_E)}\left(-d_E a_E \frac{\partial I^E}{\partial S^E}\right)\right)\right)}{\left(1 - e^{-d_E(a_E I^E - b_E)}\right)^2} \tag{2.8}$$

$$\frac{\partial r^E}{\partial S_I} = \frac{\left(a_E \frac{\partial I^E}{\partial S^I}\left(1 - e^{-d_E(a_E I^E - b_E)}\right)\right) - \left((a_E I^E - b_E)\left(-e^{-d_E(a_E I^E - b_E)}\left(-d_E a_E \frac{\partial I^E}{\partial S^I}\right)\right)\right)}{\left(1 - e^{-d_E(a_E I^E - b_E)}\right)^2} \tag{2.9}$$

$$\frac{\partial r^I}{\partial S^E} = \frac{\left(a_I \frac{\partial I^I}{\partial S^E}\left(1 - e^{-d_I(a_I I^I - b_I)}\right)\right) - \left((a_I I^I - b_I)\left(-e^{-d_I(a_I I^I - b_I)}\left(-d_I a_I \frac{\partial I^I}{\partial S^E}\right)\right)\right)}{\left(1 - e^{-d_I(a_I I^I - b_I)}\right)^2} \tag{2.10}$$

$$\frac{\partial r^I}{\partial S^I} = \frac{\left(a_I \frac{\partial I^I}{\partial S^I}\left(1 - e^{-d_I(a_I I^I - b_I)}\right)\right) - \left((a_I I^I - b_I)\left(-e^{-d_I(a_I I^I - b_I)}\left(-d_I a_I \frac{\partial I^I}{\partial S^I}\right)\right)\right)}{\left(1 - e^{-d_I(a_I I^I - b_I)}\right)^2} \tag{2.11}$$

and:

$$\frac{\partial I^E}{\partial S^E} = w_+ J_{NMDA} \tag{2.12}$$

$$\frac{\partial I^E}{\partial S^I} = -J \tag{2.13}$$

$$\frac{\partial I^I}{\partial S^E} = J_{NMDA} \tag{2.14}$$

$$\frac{\partial I^I}{\partial S^I} = -J_{new} \tag{2.15}$$

The interpretation of the eigenvalues are based on the signs and magnitudes of the real and imaginary parts, which provide information about the behaviour of the system locally around the fixed point. If the real part is positive, then a small perturbation will cause the state variable to continue to diverge away from the fixed point, whereas if it is negative it will return to the fixed point after a small perturbation. The magnitude of the real part will indicate how fast it diverges or returns. If there is an imaginary component, this means that the divergence or convergence will occur in an oscillatory fashion, where the magnitude of the imaginary part is related to the frequency of this diverging or dampening oscillation with respect to a fixed point. It should be pointed out this is determined based on the linearized version of the system, and the actual system behaviour may be vastly different after being perturbed. The eigenvalues can be categorized into six behaviours (Strang, 2014):

- Sink – Negative real numbers
- Source – Positive real numbers
- Saddle – Real numbers, one positive and one negative
- Spiral Sink – Complex numbers with negative real part
- Spiral Source – Complex numbers with positive real part
- Center - Complex numbers with zero real part

It may be of scientific interest to know how the fixed points and behaviour of the fixed points change with respect to a change in some variable of interest. In the case of NMM, we might be concerned with the change of fixed points with a change in $I_{external}$. Figures showing this visually are used in this chapter for analyzing the dynamics of the system. To generate these plots, the fixed points of S^E and S^I are calculated for a range of $I_{external}$ values, using the method described in the previous section. These plots can be computationally intensive to generate, as the method is drawing lines by plotting many individual points on the line. A very high density of points needs to be generated at the sections of the curve where there are near vertical slopes. As an extension beyond the standard analytical sensitivity plot, these points can be colored based on information from the eigenvalues of the Jacobian as described previously.

3.2.3 Analytical Oscillation Search

One may wish to find parameters such that the RWWEI model will oscillate with a certain frequency. Here we use a grid search technique, that returns parameter combinations that satisfy certain criteria. This is a brute force technique that identifies parameter sets that are predicted to oscillate within a frequency range, and it requires specifying a target mean excitatory and inhibitory firing rate.

There are two scenarios this method is based on to generate sustained oscillation in an individual node NMM. The first is limit cycle attractors (Durstewitz et al., 2022). These do not exist in linear systems, which the search problem is reduced to, but perhaps the spiral source from the linearized model will lead to a limit cycle, particularly since the state variable S^E is bounded between [0,1] in the actual equations. The second is that we may observe a peak in the power spectrum at a certain frequency if the system is a spiral sink and the system has noise so that its continually perturbed and returning at a particular frequency. That is, the noise is expected to drive the oscillations in the dampened case instead of achieving a limit cycle.

Grid search is a brute force method which works by trying many different combinations of parameters values and returning the combinations of parameters that have a desired quality. This involves defining a group of parameters, where each parameter has its own set of possible values, and then examining all combinations of these values. Brute force methods are very computationally expense, as the run time is $\Omega\left(\prod_{i=1}^{f} n_i\right)$ where f is the number of free parameters being searched and n_i is the number of possible values for the ith free parameter. This challenge reduces the number of parameters or the density of sampled points that can be searched within a reasonable timeframe. Thus, instead of running numerical simulations for all parameter combinations, an analytical method is used to predict the oscillation frequency(s), if any, that the parameter set will have.

The model parameters being searched are the four parameters regulating the local coupling and self connections of the excitatory and inhibitory populations ($J_{NMDA}, J_i, J_{new}, w_+$). This 4D search space was designed such that the values tried are above and below their default values, but constrained to not remove or reverse the direction of effect of the coupling connections.

The ranges searched were typically as follows (with step sizes of 0.05, which results in 3,672,000 combinations):

- J_{NMDA}: (0.0, 3.0]
- J_i: (0.0, 3.0]
- J_{new}: (0.0, 3.0]
- w_+: [1.0, 1.8]

Furthermore, target mean firing rates r^E and r^I are chosen, with the defaults being 3 Hz and 8Hz respectively. The corresponding S^E and S^I fixed point of target mean firing rates are calculated as follows:

$$S^E_{target} = \frac{\gamma^E r^E_{target}}{\frac{1}{\tau_E} + \gamma^E r^E_{target}} \tag{2.17}$$

$$S^I_{target} = \tau_I \gamma^I r^I_{target} \tag{2.18}$$

This results in 4 or 6 degrees of freedom being searched, depending on whether multiple combinations of target mean firing rates are being explored. For every combination of these free parameters, the input into the excitatory and inhibitory population is scaled by W_E and W_I (see Table 2.1), so these values are chosen such that the system is in the desired steady state. This is similar to how parameters were determined in (Deco et al., 2014), where I_0 and W_I were used to achieve desired firing rates.

$$W_E = (I^E_{target} - w_+ J_{NMDA} S^E_{target} + J S^I_{target})/I_0 \tag{2.19}$$

$$W_I = (I^I_{target} - J_{NMDA} S^E_{target} + J_{new} S^I_{target})/I_0 \tag{2.20}$$

Also, this is different approach from Section 3.2.1, as instead of searching for fixed points, the location of a fixed point is forced to be at the target mean S^E and S^I. That is, previously the W_E and W_I were fixed and the S^E and S^I were unknown and searched for, now the r^E and r^I of the fixed point are specified and the W_E and W_I are selected to make that so. Subsequently though, the eigenvalues of the Jacobian at the fixed points are themselves calculated the same way as in Section 3.2.2.

We have the linearized system $x' = \boldsymbol{A}x$, where $\boldsymbol{A}$ is the Jacobian of RWWEI (Equation 2.2). Then, standard solving methods can be applied, which are equivalent to finding the frequency of a pendulum or a mass on a spring:

$$Freq = \sqrt{\det(A) - \frac{1}{4}tr(A)^2} = |imag(\lambda)| \tag{2.21}$$

Converting to Hz is:

$$Freq = \sqrt{\det(A) - \frac{1}{4}tr(A)^2}\left(\frac{1000}{2\pi}\right) = |imag(\lambda)|\left(\frac{1000}{2\pi}\right) \tag{2.22}$$

When noise is present, the calculations become more complicated. There are analytical methods to estimate the power spectrum of the linearized case, and this has been applied to NMMs before (Wallace et al., 2011).

This process might return quite a few parameters, especially if a small increment is used to select values for various searched parameters. Depending on the goal of the analysis, one or more parameter sets will be used. This could be achieved by filtering, but that might add bias to the parameter sets. Alternatively, clustering could be used, where the center of a larger cluster may be a more robust parameter set.

After selecting the one or more parameter sets, it must be verified that they in fact oscillate and at the desired frequency, as though the linear approximation is an informed prediction the actual model behaviour may be quite different. Also, this behaviour may be different depending on whether noise is being added or not, and whether the node is connected within a larger network.

3.2.4 Numerical Choices and Considerations

In the results Section 3.3.2, comparisons are made between the analytically predicted dynamics and the dynamics of the system evaluated using numerical simulation. The numerical solver used is implemented in Python (NumPy) using the Euler-Maruyama method. These numerical simulations are much more computationally expensive than the analytical techniques and may require some manual trial and error. The various choices made to get good quality metrics from numerical simulation are described here.

For the chosen initial conditions of simulation, we use the fixed point (slightly perturbed in the deterministic case). If there are multiple fixed points, then the larger one is chosen. As will be seen, when there are multiple fixed points in the RWWEI model, it is observed that the larger ones are more likely to oscillate. This is much more efficient than running multiple simulations with different initial conditions. However, despite the well-chosen initial conditions, it can still be necessary to remove transient periods, during which the amplitude of oscillations may tend towards a stable value.

There are various ways to analyze the power of oscillations in a time series. One approach here that was particularly relevant to deterministic periodic oscillations was to get the amplitude as $\frac{max-min}{2}$. Then, the frequency of the oscillation can still be acquired as the frequency with the max value in the Power Spectral Density (PSD). This is based on the assumption that there is one dominant frequency in the time series, which is reasonable due to the RWWEI model being examined is a 2-state system. Bartlett's method is used to have a smoother PSD.

Often when dealing with a system that involves noise, there is a decay component of the PSD. When white noise is being integrated resulting in brown noise, a $\frac{1}{frequency^2}$ decay is visible in the power spectrum. Thus, oscillations such as alpha would be a peak on top of that line. When dealing with stochastic or non-periodic timeseries, either empirical or from numerical simulation, the power should be examined instead of the amplitude. The value from the PSD can either be the peak value in the alpha range, or the area under the curve in the alpha range. For the stochastic case, it was observed that the chosen magnitude of noise was important to have a peak in the PSD.

For the deterministic case, we have found that a small $I_{external}$ can result in oscillations that otherwise would not be present. This insight is due to the analytical sensitivity in Section 3.3.4 where the fixed points with $I_{external}$ = 0 are often at the edge of a bifurcation, where the parameter sets are produced by the proposed brute force search method. As will be seen in later analytical sensitivity plots, the found parameters are often at a bifurcation point where negative input can cause the state variables (S^E, S^I) to dampen to 0. Perhaps a small $I_{external}$ = 0.01 can be used to avoid that outcome.

3.3 Results

The results of the alpha-BOLD anticorrelation case study are presented here. First, the broader behavior of the RWWEI model in terms of the predicted oscillation frequencies from the analytical brute force search is examined. Then, with regards to the hemodynamic signal, analysis is performed which shows that steady state BOLD depends essentially on the mean S^E value for a sine wave input. Using this fact, the $I_{external}$ sensitivity analysis can also be used to look for alpha-BOLD anticorrelation in chosen parameter sets with alpha oscillations, as the steady state S^E can be used as a proxy for BOLD. Finally, two particular parameter sets are examined in detail, one with noise and the other without noise.

Longer analysis was run using the Digital Research Alliance of Canada – Advanced Research Computing resources (alliancecan.ca). This was particularly beneficial for parallelizing parameter searches. Figures were generated in Python unless specified otherwise.

3.3.1 The Parameter Space of EEG Oscillations

The parameter space of local connection strengths (J_{NMDA}, J_i, J_{new}, w_+) and input scaling factors (W_E and W_I) was searched for alpha oscillations, but it was also of interest what frequencies the RWWEI model tends to oscillate for various parameter ranges in general. As seen in Table 3.1, within the parameter ranges searched, alpha oscillations are less common in the model than higher frequency oscillations. Beta oscillations are the most common when normalizing for the size of the frequency band, but gamma oscillations have a higher occurrence of diverging oscillations. This was determined by modifying the analytical brute force search to simply count the number of oscillations at various ranges across the four coupling parameters and across r^E in [1,100] Hz and r^I in [1,100] Hz, with steps size of 1 for the firing rates.

	All	Diverging	Dampening	Normalized All (per Hz)	Normalized Diverging (per Hz)	Normalized Dampening (per Hz)
All Oscillations	16.3012%	1.0567%	15.2445%			
Delta (1-4Hz)	0.2709%	0.0058%	0.2651%	0.0903%	0.0019%	0.0884%
Theta (4-8Hz)	0.7730%	0.0179%	0.7551%	0.1933%	0.0045%	0.1888%
Alpha (8-12Hz)	1.1008%	0.0282%	1.0726%	0.2752%	0.0071%	0.2681%
Beta (12-30Hz)	6.3690%	0.2114%	6.1576%	0.3538%	0.0117%	0.3421%
Gamma (30-70Hz)	7.3968%	0.6213%	6.7755%	0.1849%	0.0155%	0.1694%

Table 3.1 – Proportion of ~37 billion parameter sets for which the RWWEI model oscillates at different frequencies.

Next in Figure 3.1, heatmaps corresponding to the first 3 columns of Table 3.1 can be seen, to determine if clusters exist. That is, whether certain combinations of firing rates of the excitatory and inhibitory populations have higher occurrence of oscillations of a particular frequency band. In fact, there are clusters for beta, alpha, theta, and delta bands when r^I has a low firing rate of less than 10 Hz. When there is also a lower r^E there is a cluster of diverging oscillations, and when r^E is high there are clusters of dampened oscillations. For theta, the occurrences of oscillations are in a more biological in terms of the r^E firing rate being below the r^I firing rate.

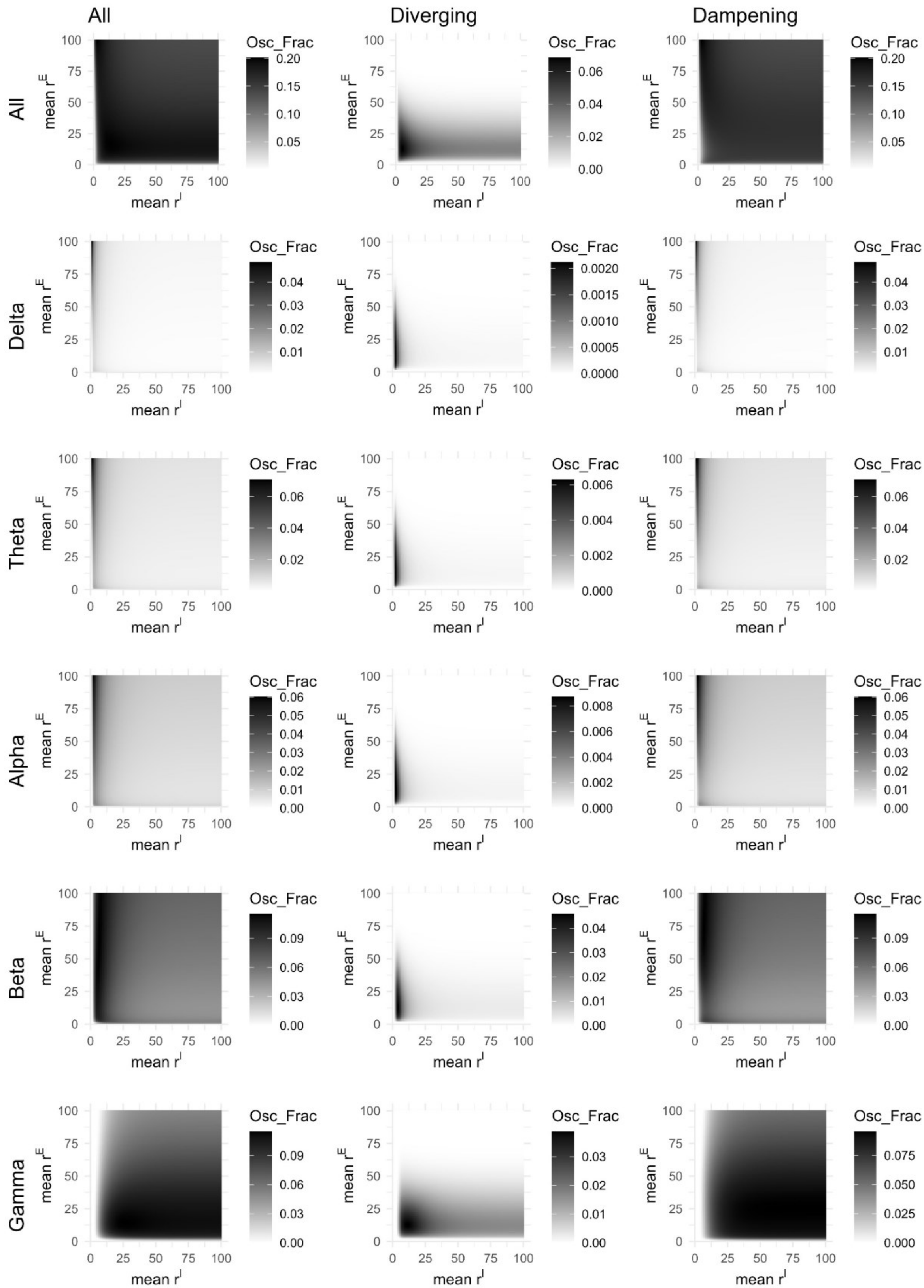

Figure 3.1 – ***Proportion of parameters that produce oscillations for different firing rates and across frequency bands***. Each pixel is a 4D parameter search, and the heatmap represents a 6D one due to different target mean r^E and r^I. (Created with R: Tidyverse)

Working with billions of parameter combinations across a six-dimensional parameter space is difficult to generalize. So, to narrow the parameter search, focuson on mean r^E = 3Hz and r^I = 8Hz is chosen which are common values used for the model (Deco et al., 2014; Wong & Wang, 2006). As a result, the problem was narrowed to millions of parameter combinations across four dimensions. At the default parameters of RWWEI, a single node does not innately oscillate. We searched around these values to try to find parameters for which the model oscillates as close to the original parameters as possible, as seen in Figure 3.2. A grid search is used to find parameters that cause a RWWEI node to oscillate near alpha. Furthermore, for each identified fixed point we calculate the Jacobian of the system at that point and look at the eigenvalues. These eigenvalues give information of the system at that point whether the real part is positive or negative (unstable or stable) and whether there is an imaginary component or not (oscillating or not oscillating).

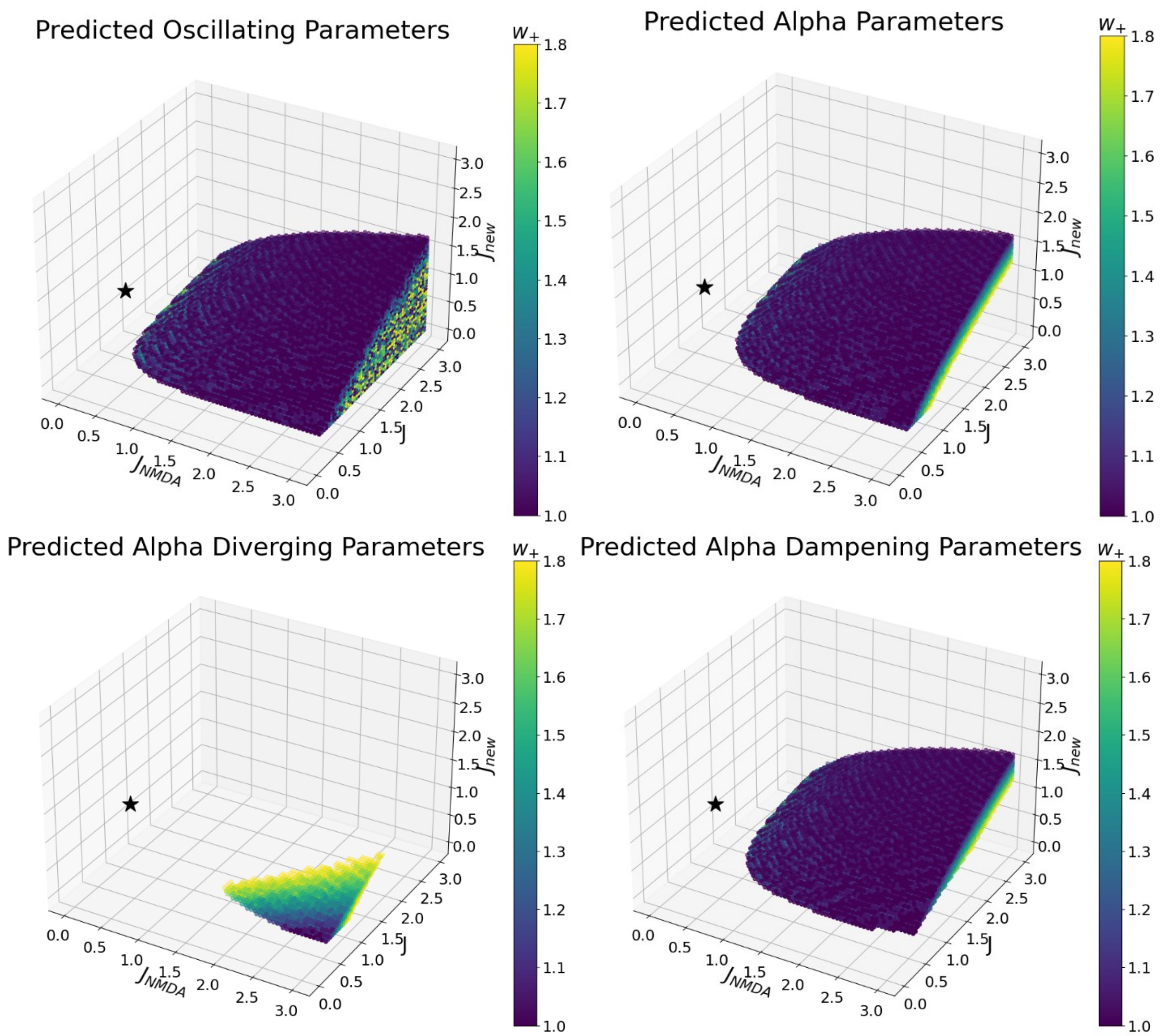

Figure 3.2 – ***Visualisation of the searched parameter space of the 4 free parameters with r^E = 3Hz and r^I = 8 Hz.*** Three of the four free parameters are on the axis, and the last one is indicated by color. The star represents the default parameters (has a w_+ value of 1.4), which does not innately oscillate without noise in the case of a single node. **Top Left:** Parameter combinations which the eigenvalue of the system predicts oscillations. **Top Right:** Parameter combinations predicted to be within alpha (8-12 Hz). **Bottom Left:** Filtered to diverging alpha. **Bottom Right:** Filtered to dampened alpha.

Due to the limited success of producing alpha oscillations from the above parameters sets, a broader search and other frequency bands are also considered. It was observed that in fact alpha occurs more frequently with higher r^E and lower r^I in the single node RWWEI (see Figure 3.1), but this is counter intuitive. Alpha is expected to be a lower energy state and also excitatory neurons have lower firing rates in general than inhibitory ones.

3.3.2 Validating the Linear Stability Analysis's Predictions

The brute force parameter search method and analytical sensitivity analysis assumes that the linearization of the model at fixed points provides information about how the true non-linear single node RWWEI model behaves. In particular, that it can provide information as to whether the model will oscillate and at what frequency. Thus, it is prudent to evaluate how well this assumption works in practice.

To validate using the linearized approximation of the system, enrichment analysis of parameters for which the model oscillates and correlations between predicted and actual frequencies is performed. A statistical evaluation is chosen because the numerical methods take much longer than that of the analytical methods, and furthermore because we are working with continuous spaces of parameter values. So, a random sample of 1000 parameter combinations with mean r^E = 3Hz and mean r^I = 8Hz is analysed for the case without noise.

These parameters are simulated with initial conditions of S^E = 0.16 and S^I = 0.08, which corresponds to mean r^E = 3Hz and mean r^I = 8Hz. They are then simulated for 60 seconds to remove the transient period. Then subsequently, another 60 second simulation was run where the criteria to be categorized as oscillating is having an S^E amplitude over the simulation measured by $\frac{min - \max}{2} \geq 0.001$.

r^E = 3Hz, r^I = 8Hz, No Noise	Linear Stability - Spiral Source (and center)	Linear Stability – Spiral Sink	Linear Stability Other	Total
Numerical Oscillation	17	1	0	18
No Oscillation	6	146	830	982
Total	23	147	830	1000

Table 3.2 – Analytical Prediction vs. Actual Oscillating tally for a random sample of parameter sets.

In the above table, the method demonstrates a statistically significant results (with $p << 0.05$ for a one-tailed Fisher Exact Test of parameter sets with an imaginary part vs. those that do not), indicating that in fact the method does work to identify when the model is naturally oscillating around its fixed points. However, to go one step further, we evaluated whether the frequency of the oscillation matches the prediction from the linear approximation.

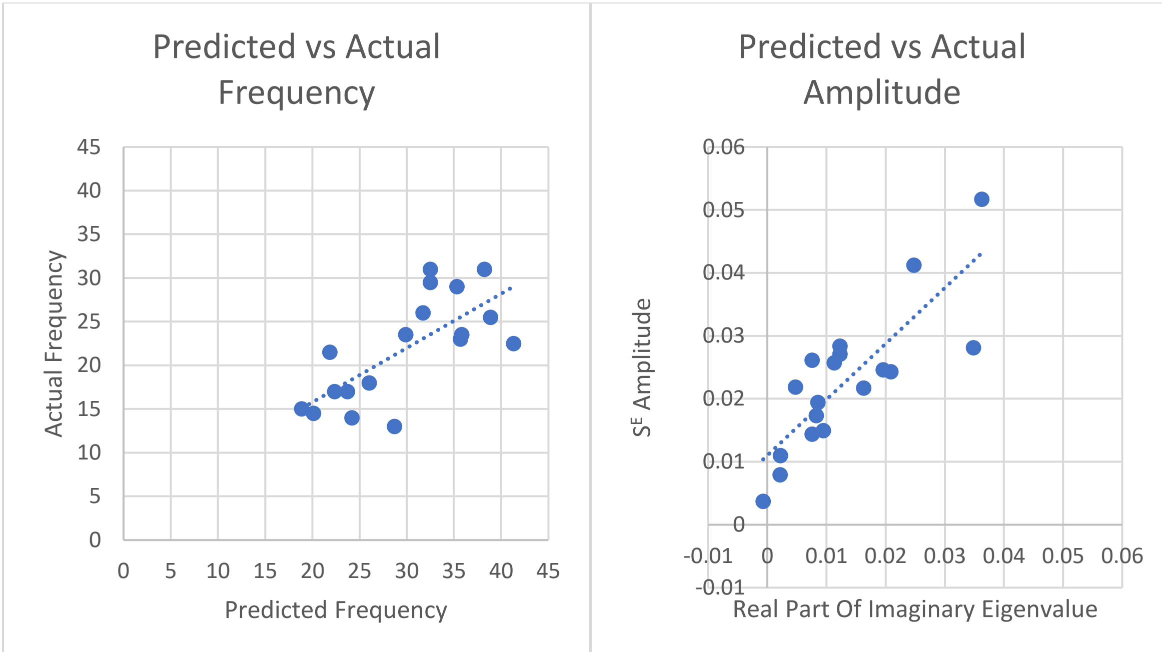

Figure 3.3 – ***Relationship between analytical estimation and numerical solutions.*** **Left:** the predicted frequency from linear stability analysis when compared to the frequency observed in numerical simulation. **Right:** Relationship between the real part of the eigenvalue with the amplitude of the numerical simulation. (Created with Excel)

We see in Figure 3.3 that there is a strong correlation between the predicted and actual frequency, though the actual frequency tends to be slightly lower than predicted. This concludes the validation of the case of the deterministic single node NMM, for changing local coupling parameters with mean r^E = 3Hz and mean r^I = 8Hz. This is quite useful, because the analytical techniques are much less computationally intensive than numerical methods and do not require tuning, for which different choices could have different results. Numerical simulation can then be useful to analyze with higher accuracy once the parameter sets have been narrowed down.

This would be interesting to do for the case of noise as well, but it is a more challenging problem as amplitude would no longer be used to measure whether oscillations are present. Instead, one would need to look at whether there is periodic bump in the above the aperiodic component of the PSD (Donoghue et al., 2020).

3.3.3 Analysis of the BOLD Equations

Now that we have EEG oscillations, we turn to analysis on the BOLD equations. As it turns out, the steady state of the BOLD equations (Table 2.3) can be explicitly solved. These are the equations of the steady state, which depend only on a constant input z_i, for which S^E from RWWEI is used:

$$x_i^{ss} = 0 \tag{2.23}$$

$$f_i^{ss} = \frac{z_i - \gamma}{\gamma} \tag{2.24}$$

$$v_i^{ss} = \left(\frac{z_i + \gamma}{\gamma}\right)^{\alpha} \tag{2.25}$$

$$q_i^{ss} = \frac{1}{\rho}\left(\frac{z_i + \gamma}{\gamma}\right)^{\alpha}\left[1 - (1-\rho)^{\left(\frac{\gamma}{z_i + \gamma}\right)}\right] \tag{2.26}$$

We investigate the response of the BOLD dynamics to various properties of input signal, to see whether these equations themselves can produce an alpha-BOLD anticorrelation. Using numerical simulation we determined that the steady state BOLD depends essentially only on the mean value of S^E, when assuming a sinusoidal input. Three experiments were performed where a sinusoidal input was injected into the model with different frequencies with the same amplitude and mean, different amplitude with the same frequency and mean, and different mean with the same frequency and amplitude. The change in mean had a much larger effect than the other two, as seen in Figure 3.4.

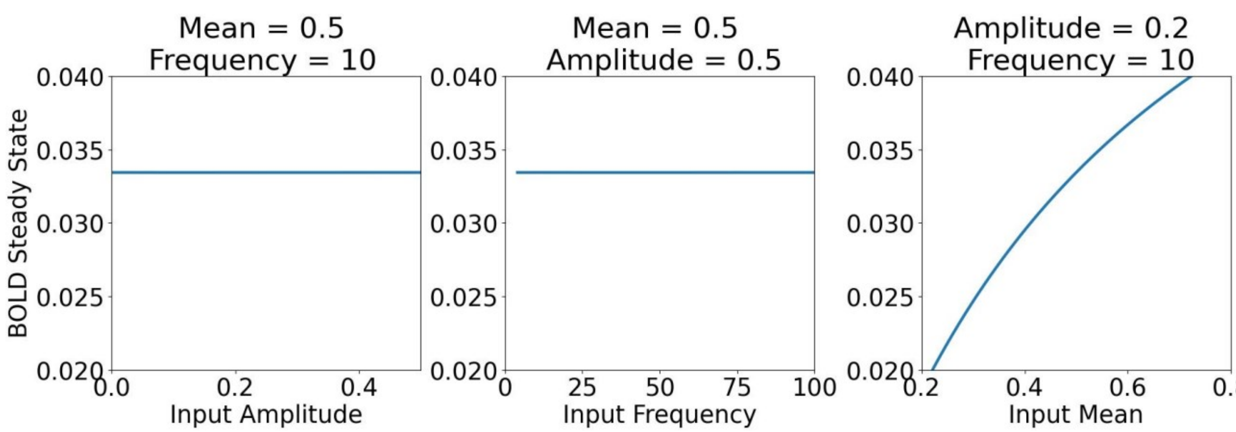

Figure 3.4 – ***BOLD Steady State with respect to changes in the amplitude, frequency, and mean of a sine wave input*.** The BOLD Steady State depends essentialy on the input mean. Each plot represents a line within a 3D search space (Amplitude, Frequency, Mean), which are orthoginal search directions. A step size of 0.001 seconds was used for simulations.

This observation justifies simplifying the alpha-BOLD question to an alpha power vs. mean S^E problem, ignoring the extra BOLD dynamics, as mean S^E can be used as a proxy for BOLD. This will be helpful in interpreting the fixed point analysis in the next section, as the BOLD value can be inferred based on the mean S^E value alone. However, these results assume a sinusoidal oscillation, where the time spent above and below the mean cancel out. Also, it assumes there are no very low frequency (< 1Hz) oscillations which are on a similar time scale to the BOLD dynamics.

3.3.4 External Input Sensitivity Analysis

Based on Section 3.3.2 and Section 3.3.3, it now follows that an analytical sensitivity analysis on $I_{external}$ can be used to study the alpha-BOLD anticorrelation phenomena. In particular, determining how changes in $I_{external}$ influences the mean, amplitude, and frequency of the NMM state variable S^E. The S^E fluctuations are a proxy for EEG oscillations, and the steady state S^E is a proxy for the BOLD signal. Furthermore, in the case of an isolated node which is being studied, $I_{external}$ is being interpreted to be the aggregation of long-range inputs from the other areas of the brain. Thus, $I_{external}$ changes can indicate whether the brain is entering a higher or lower active state. In doing so, associations between frequency and power of oscillations with BOLD can be examined.

We can calculate the magnitude of the expected amount of input change a single node could receive from the network. First, noting that the firing rate of r^E of 3Hz and r^I of 8Hz corresponds to a mean S^E ~= 0.16 and mean S^I ~= 0.08. Then, noting that Deco et al., 2014 had tested ranges of G on the order of 0-4 with the FIC method. Thus, we can expect that the single node should be able to receive input variation on the magnitude of 0.1 or 1.0 nA's, though there is some ambiguity due to the connectivity matrix C_{ij} and its normalization.

There are two parts to the $I_{external}$ sensitivity analysis. The first is to select a random set of parameters predicted to be oscillating with mean r^E of 3Hz and r^I of 8H in order to observe the variation in behavior. That is, in terms of the variation in oscillatory properties and in steady state S^E (the proxy for BOLD). This was performed to give some context for the two particular parameter sets subsequently examined to come up with candidate mechanisms for alpha-BOLD anticorrelation.

3.3.4.1 Variation Observed in Oscillatory and Steady State Properties

Analysis was performed on a random sample of 50 parameter sets for which the model oscillated that had biologically realistic mean firing rates of r^E = 3Hz and r^I = 8 Hz (Brunel & Wang, 2001; Wilson et al., 1994). These parameter sets are analyzed initially with analytical methods (Figure 3.5) and then further verified with numerical methods (Figure 3.6). From the numerical simulations, the deterministic case looks directly at amplitude, but for the stochastic case power is examined. The goal is to observe the variability in the change of oscillation characteristics, to see what is common across parameter sets for which the model oscillates. The range of $I_{external}$ of [-2,5] is chosen to observe the full range of S^E, however, noting that this has been seen to cause the S^I to go much beyond the biologically relevant range of between 0 and 1. Furthermore, in a network setting the range of $I_{external}$ which is being used to represent input from the network is more likely to be between 0 to 1, but this depends on the connectivity matrix and global coupling scaling.

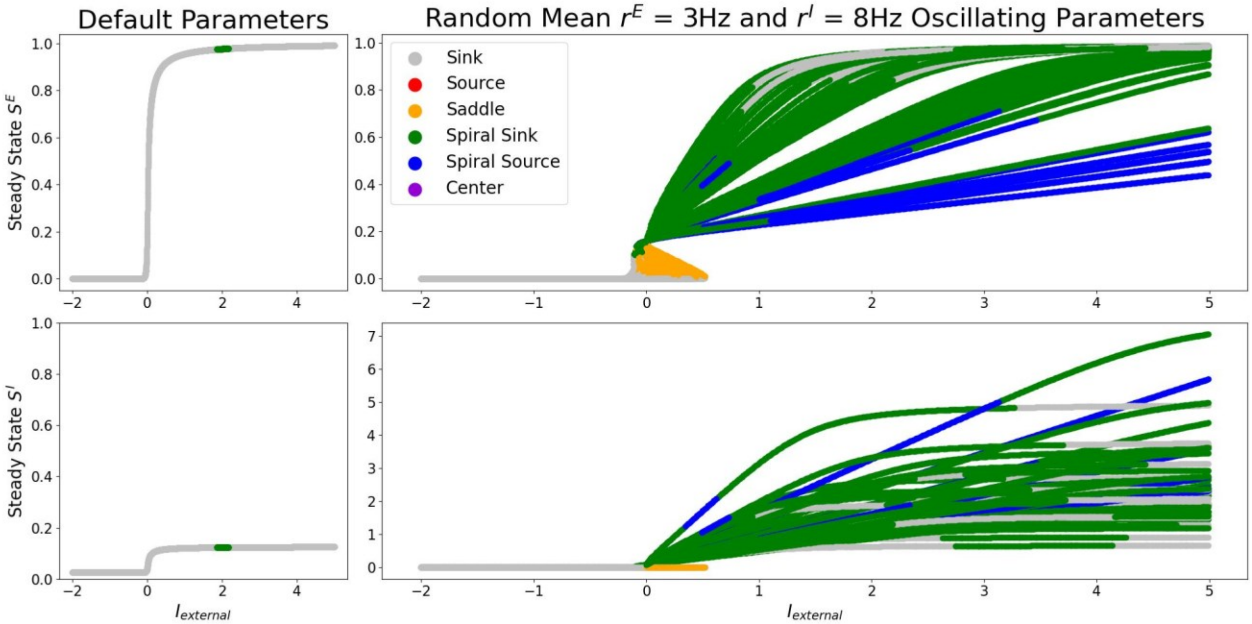

Figure 3.5 – ***Analytical sensitivity of an individual RWWEI node.*** **Left:** Analysis of default parameters. **Right:** 50 example parameter sets for which the model oscillates around r^E = 3Hz and r^I = 8Hz. Variation in the steady state values and in ranges of dampened vs diverging oscillating predictions is observed.

There are various structures seen in the analytical sensitivity plots, in Figure 3.5, for the 50 randomly selected parameter sets. Most parameter sets have a hysteresis, though not all. Also, many parameter sets have hopf and reverse hopf bifurcations. A similar analysis has been performed on Wilson-Cowen model with the same hysteresis and hopf bifurcation features observed (Negahbani et al., 2015). There appears to be some degeneracy in the two-state models, due to the similarity across RWWEI and Wilson-Cowan, in terms of the bifurcation structures for oscillating models.

In all of these parameter sets, we observe that an increase in $I_{external}$ results in an increase in mean S^E, as would be expected. Furthermore, there is variability in the range of hysteresis and the range of unstable oscillations. However, it was unexpected to find that for the random parameter sets the hysteresis usually starts around $I_{external}$ = 0 and oscillatory regions disappear with negative $I_{external}$. When trying other mean firing rates, more variability of where hysteresis started was observed, though it was still often around $I_{external}$ = 0.

The subsequent numerical sensitivity plots, in Figure 3.6, give information about the range for which oscillations occur, and the associated frequency and power of the oscillations. Consistent with what was previously observed, most oscillations are at frequencies higher than alpha, and alpha does not appear to be robustly preserved. Interestingly, the stability of the system at $I_{external}$ = 0 is a strong indicator as to whether oscillations will occur at all if noise is not added to the system. Oscillations are more robust with higher instability. Also, the range of input for which oscillations are present is extended when there is noise in the system.

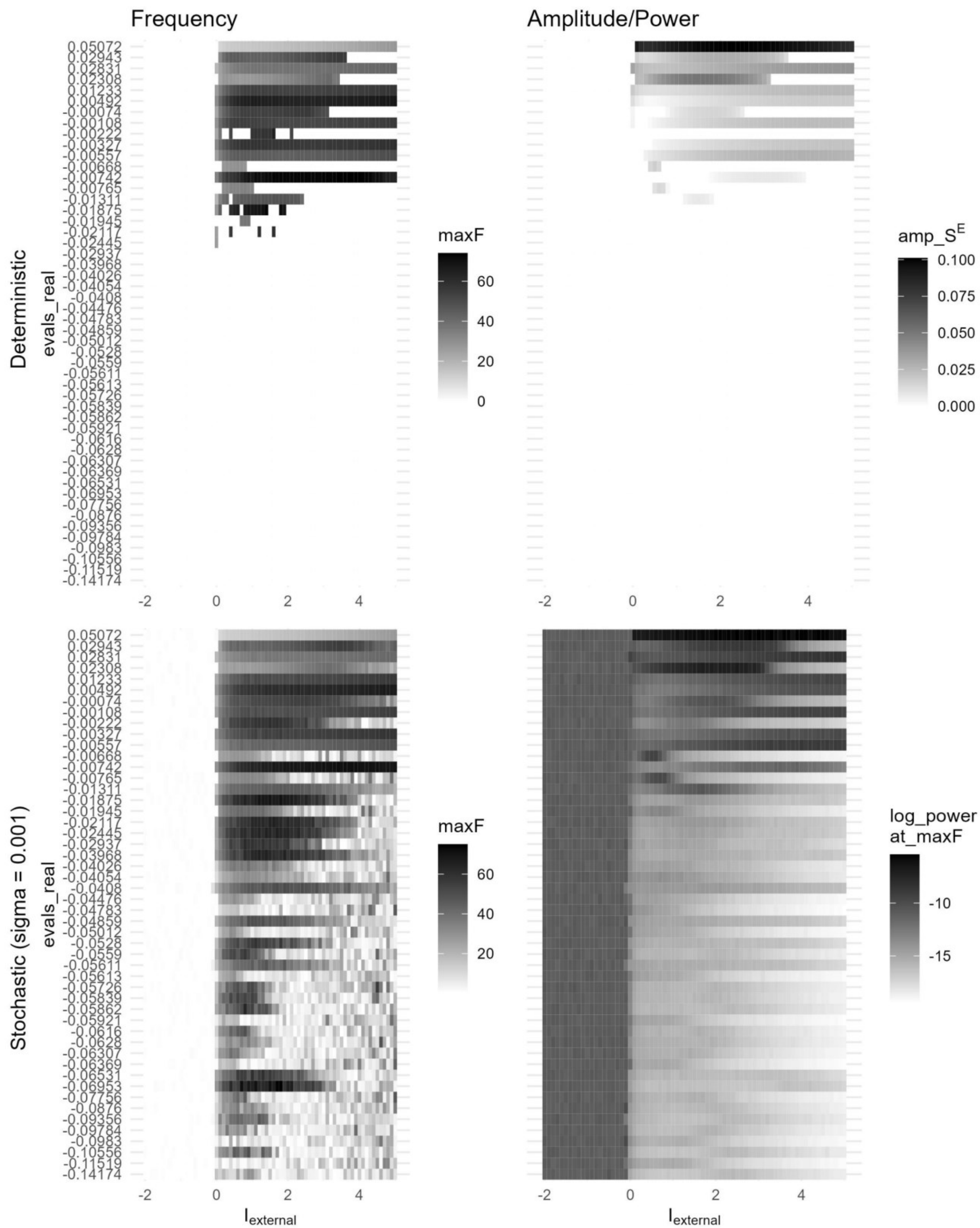

Figure 3.6 - *Numerical sensitivity analyses for 50 example parameter sets for which the model oscillates.* These parameters sets were predicted to have r^E = 3Hz and r^I = 8Hz at $I_{external}$ = 0. **Top:** simulations run without noise. **Bottom:** simulations run with noise. The parameter sets are sorted along the y-axis according to the real part of their eigenvalues, which represents stronger divergence to stronger dampening from top to bottom. (Created with R: Tidyverse)

3.3.4.2 Parameter sets to study the Alpha-BOLD Anticorrelation Phenomenon

Next, a closer look is taken for two chosen parameter sets to study the alpha-BOLD anticorrelation phenomenon. Ideally, these parameters would have been found by fitting the model to an empirical multimodal dataset. However, for the purposes of this case study, it is the parameter set found analytically, filtered to those with a real part between +/- 0.001 and an imaginary part indicating 9.5-10.5Hz, and with the smallest Euclidean distance to the original parameter values for the 6 parameters (J_{NMDA}, J_i, J_{new}, w_+, W_E, W_I), such that when simulated, an innate oscillation is observed. This is noting that, there will likely be a bias for the Euclidean measurement as the parameters are not all of the same units or of the same importance, but this is not problematic as the actual values themselves are not particularly important in this case study. Regardless, the values chosen by the described process are as follows:

Chosen RWWEI Parameters for which Alpha Oscillating Occur	
Dampened (Stochastic)	Diverging (Deterministic)
J_{NMDA} = 1.2 J_i = 1.05 J_{new} = 0.05 w_+ = 1.8 W_E = 0.293607 W_I = 0.223681 σ = 0.0005 $I_{external}$ = 0.00	J_{NMDA} = 1.45 J_i = 1.00 J_{new} = 0.05 w_+ = 1.8 W_E = 0.093141 W_I = 0.118128 σ = 0.00 $I_{external}$ = 0.002

Table 3.3 – Chosen parameter sets for which alpha oscillations occur (dampened and diverging cases).

In both the diverging and dampened diagrams in Figure 3.7 and Figure 3.8, we see consistent results for these parameter sets that were predicted based on the previous sections. The dampening parameter set has a very similar analytical sensitivity structure to the diverging parameters. The main difference being that the system has a region of dampened oscillations before diverging oscillations occur. As expected, the range of input for which oscillations occur in numerical simulation is extended into the dampened regions by noise, though the power is much lower there.

The change in steady state S^E indicates the change in BOLD, where the imaginary part of the eigenvalue at the fixed point indicates frequency, and the real part the amplitude. We see that it is common for the oscillation frequency and amplitude to change at the same time. So, alpha seems to exist only in a very small range.

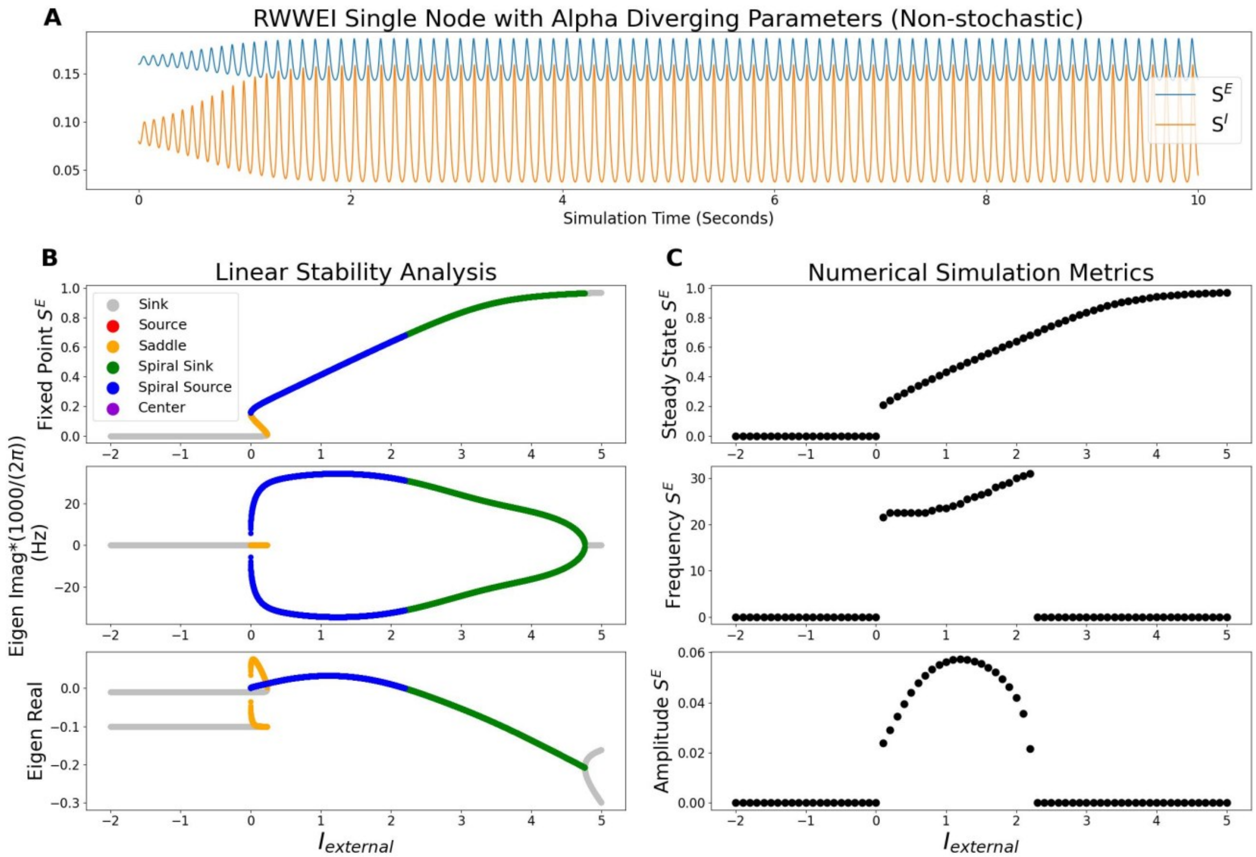

Figure 3.7 - ***Analytical and numerical sensitivity analysis of the RWWEI with deterministic alpha oscillations to changes in $I_{external}$.*** **Panel A:** oscillating time series that occurs near $I_{external} = 0$. **Panel B:** the analytical sensitivity. **Panel C:** the numerical sensitivity. Initial conditions of numerical simulations were based on the larger fixed point.

Multiple properties of the chosen parameter sets were sensitive to $I_{external}$ and the noise magnitude. Despite filtering to mean r^E = 3Hz and r^I = 8Hz, when adding input firing rates quickly left those values as seen in Figure 3.7 and Figure 3.8. Additionally, there is variability in the range of diverging oscillations that occur. Correspondingly, in the metrics from numerical simulation in Figure 3.6, variability in the frequency and amplitude/power is also seen. It appears that not only are beta and theta oscillations more common, but they were also more robust to changes in input.

This made the model difficult to work with in terms of looking for an alpha-BOLD relationship, because a parameter set was not found where the alpha frequency was robustly maintained while having the amplitude or power show a smooth change. The found parameter sets for which alpha oscillatons occur are very sensitive to changes in net brain activity and noise. Negative inputs can stop the oscillations entirely and the model could get stuck at the bottom of the hystereisis. Positive inputs can rapidly shift the oscillation to higher more robust oscillation frequencies. Furthermore, at these higher frequencies the inhibitory population is often outside of the biological range. Finally, to complicate things further, different ranges of $I_{external}$ have different effects on the oscillations in terms of amplitude and frequency.

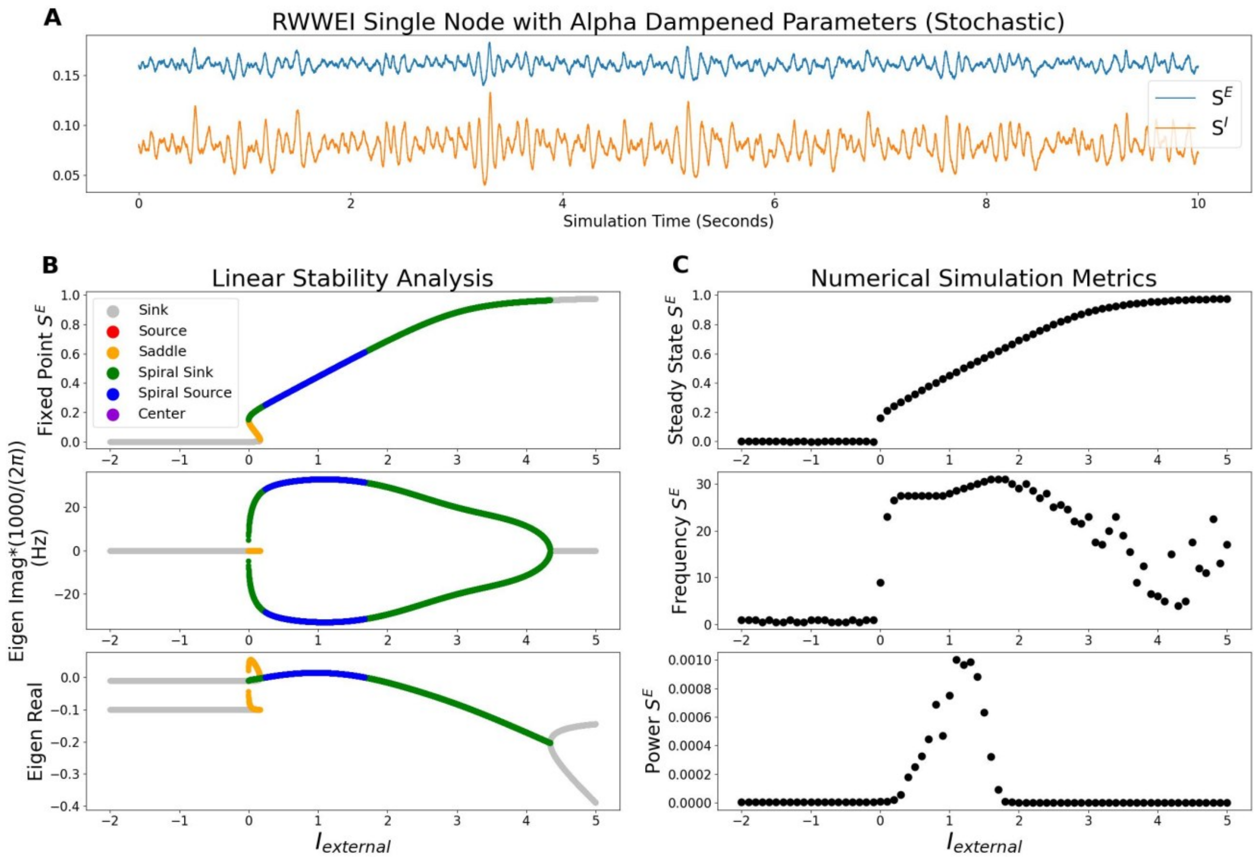

Figure 3.8 - ***Analytical and numerical sensitivity analysis of the RWWEI with stochastic alpha oscillations to changes in $I_{external}$.***
Panel A: oscillating time series that occurs near $I_{external} = 0$. **Panel B:** the analytical sensitivity. **Panel C:** the numerical sensitivity. Initial conditions of numerical simulations were based on the larger fixed point.

3.4 Discussion

This case study investigated a multimodal NMM model using linear stability analysis and numerical techniques. This involved both optimizing NMM parameters to produce certain oscillatory characteristics, as well as analyzing the model within the found parameter ranges. Here the interpretation of the results regarding the alpha-BOLD debate are discussed, along with the various limitations and future work. Within this case study, multiple methods and techniques have been presented which will also be relevant for later chapters. Challenges of using brute force and linear stability analysis techniques alone are mentioned. Then, extensions of how they can be complementary to gradient descent parameter fitting methodologies are discussed.

3.4.1 The Alpha-BOLD Anticorrelation Question

3.4.1.1 Two Candidate Mechanisms

While ideal biological parameter sets satisfying all the specified characteristics of alpha power vs. BOLD anticorrelation in a single node NMM were not found, potential mechanisms can still be hypothesized. As these parameter sets were not found by fitting to empirical data, we are more interested in the underlying dynamical structure that could result in an alpha-BOLD anticorrelation or an oscillatory power vs BOLD anticorrelation in general. This is done by looking at the structure of the system for parameters sets that partially satisfy the set of alpha-BOLD characteristics, at least over certain $I_{external}$ ranges, which represent the net input from the rest of the brain at different activity levels. It is possible a more extensive search could find parameter sets that maintain the existing properties while also achieving the missing ones.

Candidate mechanisms can be hypothesized by examining the sensitivity analysis presented in Figure 3.7 and Figure 3.8. This is because the initial results shown in Figure 3.4 imply that the alpha-BOLD anticorrelation in the RWWEI model could be simplified to looking for mechanisms that cause an alpha vs. mean S^E anticorrelation. As a result, this analysis predicts the change in oscillation frequency and power as a result of changing overall brain activity level. When finding self-sustained oscillation without noise the alpha power and BOLD remained constant due to the simplicity of the dynamics. Thus, changes were induced by running steady state simulations with a set of constant input currents. As would be expected, a monotonic increase in S^E activity was observed when increasing $I_{external}$, for which the steady state was interpreted to represent a proxy for BOLD. So, for the various parameter sets, the NMM was put into a higher or lower steady state BOLD states via the $I_{external}$ input that represented the activity from the rest of the brain.

The first candidate mechanism for the alpha-BOLD phenomenon is a shift in the frequency from alpha to higher frequency bands. The idea of alpha shifting is an existing idea in the literature (Schirner & Ritter, 2023). As the found alpha oscillations in RWWEI are very sensitive, the alpha power may decrease with increased $I_{external}$ as the oscillation frequency rapidly increases to the beta or gamma ranges. This is consistent with the idea that higher alpha power is a lower energy state. Thus, for a very narrow range there is a high alpha power, and everywhere else alpha power is very low. If the lower bound of input to the node was put near that alpha oscillating area, and the model does not unrecoverably loose oscillations at the left point of the hysteresis where the alpha oscillations are very near to, then a statistical anticorrelation would be expected to be observed. This proposed mechanisms of the shift to higher frequencies when BOLD increases is similar to the argument presented by Pang & Robinson.

The second candidate mechanisms is that an alpha-BOLD anticorrelation can be produced by passing through a reverse hopf bifurcation where diverging oscillations become dampened oscillations. In order to see this in found parameters, two problem constraints must be relaxed. First, the alpha-BOLD anticorrelation must be relaxed to looking for an oscillation power vs BOLD anticorrelation. Second, the biological target mean firing rates must be relaxed to allow for much higher mean firing rates. Then, an oscillation power vs. BOLD anticorrelation can be found. For lower $I_{external}$ a positive correlation between the oscillation amplitude vs. BOLD is in fact observed. However, for higher $I_{external}$ ranges that are no longer biological mean firing rates, a beta-BOLD anticorrelation occurs in the model explained by a reverse hopf bifurcation.

3.4.1.2 Limitations

There are various limitations to the approach used to study the alpha-BOLD phenomenon, beyond the relaxation of biological constraints already discussed. Firstly, this case study approaches the alpha-BOLD question from a theoretical perspective that is based on multiple simplifying assumptions. The RWWEI model itself is simplified to have single differential equations to represent a whole population of neurons. Additionally, there are some inherent assumptions about alpha-BOLD anticorrelation itself due to the choice of NMM and the input used to study the phenomenon.

An assumption was made that the alpha-BOLD phenomenon is something that can occur as a result of local brain region activity. In fact, this may be signalled from other areas of the brain, or a result of interactions between brain regions. It is expected that network level modelling could also come up with candidate mechanisms for alpha–BOLD anticorrelations. Perhaps in a network the alpha peak and alpha-BOLD anticorrelation are separate phenomena such that an alpha peak might not be necessary for the alpha-BOLD anticorrelation to exist.

Choosing S^E state variable to represent source space EEG has multiple caveats. This variable is about overall synaptic activity instead of capturing neural synchrony, which the EEG signal is believed to be a result of. Next-generation NMMs have been created to address this issue (Byrne et al., 2020), but may be more difficult to analyze analytically. Also of note, the S^E state variable is bounded by a range of 0 to 1 (though noise can perturb it out of the range), so a sinusoidal type of oscillation would be expected to have the maximum possible oscillation amplitude at mean $S^E = 0.5$ and would decrease when the mean either increases or decreases. This is consistent with what is seen with the amplitude vs. $I_{external}$ results, but the effect of these boundaries, which may also contribute to the oscillations, do not necessarily produce biologically meaningful candidate mechanisms.

The simplification of using steady state S^E as a proxy for BOLD in analysis assumes that the mean of the oscillations is the steady state value itself. However, given the nonlinearity of the model this might not be true. Additionally, the current results looked at 3 orthogonal lines in 3D space, so a more rigorous exploration could be performed. In fact, an alpha-BOLD anticorrelation could result from changes in the oscillatory shape which are not a sine wave. For example, if the wave stayed in the trough longer than in the peak, then increased amplitude would cause the mean to decrease resulting in an anticorrelation. In fact, this idea is related to the explanation for the alpha-BOLD anticorrelation in Schirner et al. (2018).

3.4.1.3 Future Work

The analysis of this case study could be extended in multiple ways. Firstly, other parameters, or broader parameter ranges could be explored. Some additional analysis on RWWEI could be done, but it would also be useful to see if there are different or common mechanisms found in other NMM. In fact, for looking at an individual brain region higher resolution models that look at individual neurons may be of interest. Finally, working with concurrent EEG and fMRI datasets could be helpful to narrow down the mechanisms being examined.

More exploration of the RWWEI model parameter space could be performed. Perhaps the parameter search performed was too narrow and could have included more parameters over larger ranges. The focus was on the local coupling strengths, but there are many other model parameters including time constants and firing rate equation parameters. Thus, the full range of model behaviours may not have been observed. Additionally, to simplify the problem, the results were often constrained to focus on r^E = 3 Hz and r^I = 8 Hz, though results show that alpha is less present for these firing rates. Another choice that was made was to have external input into the excitatory population, but not into the inhibitory population. This choice was biologically motivated, but it may be a simplification. Perhaps in a 2D sensitivity analysis for input into Excitatory and Inhibitory populations, by changing W_E and W_I, a ratio could be found that causes an alpha power vs BOLD anticorrelation.

Similar exploration could be done in different NMMs. It has been noted that EEG is better at predicting fMRI than fMRI is as predicting EEG (Deligianni et al., 2014). Thus, it might make more sense to use a model which has been commonly used for EEG, than to use the RWWEI model that has been commonly used for fMRI, when trying to generate concurrent EEG and fMRI timeseries. There might also be a better dynamic to use as a proxy for source space EEG than S^E. RWWEI model has a strong inhibitory behavior, compared to other models like Wilson-Cowan, as it does not have a (1 - S^I) term to bound S^I below 1. Alternatively, for analysis only involving one brain region, it may make more sense to work at a higher resolution model of individual spiking neurons directly.

Analysis could also be performed on the network level. It would be valuable to compare parameters which had a positive correlation with those that have an anticorrelation to hypothesize its existance. This could be done in the form of fitting local node parameters in a network to empirical data, as some brain regions show a positive correlation while others show an negative correlation.

A fundamental thing missing from the case study of this chapter is replicating the alpha-BOLD anticorrelation analysis in a concurrent EEG and fMRI dataset. Analysis on such a dataset could help narrow down the candidate dynamical systems mechanisms. It could also give an indication as to how well the model generally matches the data, such as how robust alpha is for various ranges of BOLD. Furthermore, we are not observing alpha-BOLD in a particular dataset and not fitting the model to a empirical dataset. Instead, an abstracted approach of looking for mechanisms in dynamical systems was used. But it would be good to be able to fit the model to particular multimodal empirical recordings, and that is what later chapters of this thesis will be about.

3.4.2 Relevance of Case Study Methods for Model Fitting

There are various ways that the techniques presented in the case study of this chapter can be extended and used for CNMM parameter fitting and its evaluation in general. Linear Stability Analysis can be used for finding initial model parameters which can be further tuned by methods presented in later chapters. Alternatively, objective function components could be created based on linear stability analysis metrics, for use in other optimization techniques such as gradient descent through backpropagation. Finally, using linear stability analysis to find local optimal parameter values can be used to evaluate the performance of gradient descent parameter fitting methods.

After model parameters have been fit, they can be compared across the brain regions of individuals or globally across subjects. An example of such analysis would be to look at the analytical sensitivity plots for varying J_i (and $I_{external}$ to maintain r^E) to simulate FIC method and look at how the bifurcation or oscillation ranges change with a change in J_i. Analytical sensitivities or bifurcation points between cases and controls can also be compared when fitting the model to each subject individually. Alternatively, comparing the range of input for which hysteresis is observed, which is related to working memory for decision making (Wong Wang, 2006). These techniques can be used to study many different biological questions.

3.4.2.1 Motivation for going beyond brute force parameter searching methods

There is a practical limit for how many parameters can be searched in a grid search, as the run time generally increases exponentially for every parameter that is being added in brute force methods. The limitation is for finding biologically relevant parameters, and in the number of parameters that can be densely searched with brute force. Furthermore, the parameter sets found, if they exist, are of the form of multi-dimensional subspaces. Generalizing these subspaces or choosing the ideal parameter set from the space is quite challenging and time consuming. This leads to the motivation of later chapters, which can be used as a method to select an individual parameter set based on fitting to a dataset.

3.4.2.2 Combining Linear Stability Analysis with Gradient Descent Methods

The oscillation search method presented in this chapter focused on adjusting a particular set of model parameters, local coupling values and base input values. This method could be altered in a straightforward way to fit other parameters of the NMM, such as the time constants or the parameters of the firing rate equations. Perhaps the constraint of particular firing rates could also be removed for a broader search. This analysis has focused on RWWEI model, but it could be applied to a different NMMs as well.

One may wish to find parameter sets as initial conditions to be further tuned by a gradient descent based technique (as is done in Section 5.3.3). Additional characteristics such as the range of input for which hysteresis is present or the range of input for which divergent oscillations occur may also be combined with the grid search to have more specialised parameters. The mean and variance of a characteristic across input values could be used in a criteria for robustness, particularly related to oscillation characteristics. Using information from the analytical fixed point sensitivity as part of the brute force search itself, is a fast way to further narrow down parameter combinations compared to using numerical methods, and could be particularly useful for further alpha-BOLD searches. Overall, this saves a lot of computational time because far fewer numerical simulations have to be run.

The methods used here can be integrated into an objective function components in various ways. The techniques for analyzing time series can be used to compare simulated and empirical data in an objective function to fit to the data. Alternatively, analytical methods such as estimating frequency based on the system's Jacobian (the method which avoids the use of imaginary values) can be added directly to an objective function, independent of empirical data. For example, the equations in Section 3.2.2 and Section 3.2.3 could be integrated into a loss function.

3.4.2.3 Evaluating Gradient Descent Methods

The properties of the model over a 1D or higher dimensional parameter space can be explored using linear stability analysis combined with a grid search to determine whether a nearby local optimum parameter set exists. This is useful to evaluate whether a gradient descent algorithm is able to successfully find the known local optimum (as is used in Section 5.3.1).

CHAPTER 4 – PYTHON PACKAGE: WHOLE BRAIN MODELLING IN PYTORCH

Abstract

Whole Brain Modelling in PyTorch (WhoBPyT) is a python package developed to generalize the parameter fitting methodology presented and used in two papers from the WBMG. In one paper a RWWEI CNMM was fit to resting-state fMRI (Griffiths et al., 2022), and in another paper a Jansen-Rit CNMM was fit to evoked potential EEG (Momi et al., 2023). This chapter highlights the software engineering aspect of the project, regarding contributions to the code architecture and support for various combinations of models, objective functions, and parameter fitting paradigms.

We start by positioning the WhoBPyT methodology within the literature. The core idea of the deep learning technique, used by WhoBPyT, is to implement CNMM simulators (DE Solvers) in PyTorch to generate synthetic neuroimaging data. In doing so, the simulated output can be backpropagated through time to update the model's parameter values. This can be interpreted as the CNMM itself being a custom Continuous Time Recurrent Neural Network (CTRNN). The relationship comes from the similarity between the forward pass of an RNN and a Euler-Maruyama simulation, particularly for network-based dynamical systems. To generate the error metric for backpropagation, an objective function based on the simulated output is used. Gradient descent can then be performed to optimize parameters.

Subsequently, we describe the package architecture and built-in functionality. Abstract classes were defined to help bring uniformity to the API. Many features used in the original publications were maintained as options through method arguments to bring more transparency to the approach. That said, some built-in or optional modifications to the implementation may be included for the purposes the optimization, numerical stability, or to reduce the computational requirements. Thus, evaluating the found parameters in a new verification implementation or using other CNMM implementation, is highly recommended, and the architecture is designed to make this task more convenient.

A new implementation of the RWWEI model (class named RWWEI2) which follows the original model equations more strictly is presented. This includes new features that have been added as part of this thesis work to support fitting to concurrent resting-state EEG and fMRI as well. This is used to test out a new feature of fitting local NMM parameters. Some new fitting paradigms and objective functions are defined and implemented, for the purposes of working towards concurrent resting state EEG and fMRI that is evaluated in the next chapter. Batching and GPU support was introduced for these new classes. A subset of these new classes and features are then used in a simple demonstration.

Finally, the chapter concludes with a summary of the current state of the package and a discussion of future extensions that could be made.

4.1 Introduction

Whole Brain Modelling in PyTorch (WhoBPyT) is a Python package started in the WBMG and based on the work of John Griffiths and Zheng Wang with RWWEI for resting-state fMRI, and additionally applied to Jansen-Rit for evoked potential EEG by Davide Momi. In this introduction the WhoBPyT Python package and methodology is positioned within the literature.

4.1.1 CNMM DE Solvers as Continuous Time Recurrent Neural Networks

CNMMs and RNNs can both be used to generate neuroimaging time series. In our case, we can furthermore implement a CNMM DE solver so that it is equivalent to the forward pass of a *Continuous Time Recurrent Neural Network* (CTRNN). This means that if we use the CNMM as a CTRNN, then the model we are directly training will be more constrained, and likewise more interpretable biologically, than would be the case for a generic RNN. The extra biological interpretability is a tradeoff with the extra difficulty for a gradient descent via BPTT optimization algorithm to work.

There is a natural correspondence between the CNMM parameters and state variables and the RNN parameters and hidden states. The state variables of the CNMM differential equations would be interpreted as the hidden states of the custom CTRNN and when unrolled would represent neural activity time series. For example, in the case of RWWEI as an Elman RNN (see Table 2.5; Elman, 1990), the dimension of h_t could be number of brain regions multiplied by the number of state variables per NMM node. Furthermore, for resting-state data, the W matrix could be the identity matrix, with the x_t representing noise. The fitted parameters represent the parameter of the CNMM equations. If a neuroimaging modality such as BOLD dynamics are added, this can be considered a second custom RNN layer of a different kind. BPTT can then be performed through both.

For the CNMM DE solver to be equivalent to an CTRNN forward pass, we use a first order estimation method. This solver would be Euler's method for deterministic equations and the Euler-Maruyama method for stochastic differential equations (see Section 2.4). In both cases, a recurrent algorithm uses the current state to calculate the next time point, without needing to know the previous history of the system. A difference, however, is that CNMMs have more complicated equations for the nodes than typically would be used in a CTRNN. That said, an alternate interpretation of the CNMM as a CTRNN formulation could be to treat each local neural population in the NMM as a separate node, with a resulting larger network but where some connectivity matrix values will be from the SC and others will be from the local NMM connection strengths.

Despite similarities in the interpretation of the forward pass, CNMMs can be more challenging to train in general. RNNs are more often used in machine learning where problems are approached by selecting the model with the best performance, after training models of various architectures and trying various hyper parameters. However, in the case of a CNMM, certain parameters and hyper-parameters are constrained by biological measurements, or to ensure valid numerical integration. Due to these constraints, where the representation of nodes and the meaning of parameters are fixed, it is likely a more challenging problem to optimize, which needs to be convex in order to succeed. This is different from regular ML problems where the trained weights could form abstract representations that change over training reducing the need for global convexity.

In this thesis, we focused on using CTRNNs to optimize parameters of personalized models to resting-state neuroimaging data, though perhaps in the future a more standard ML approach (splitting the empirical data into training, validation, and testing sets) could be used. For now, the goal is to develop and evaluate the training algorithm's ability to fit a model to neuroimaging phenomena from one recording session as closely as possible. In this case, the concept of overfitting from the ML perspective is not relevant, as generalizability across subjects is not being claimed, though we wish to work towards that generalizing ability. Thus, although we are using an RNN technique, for the present purposes we are essentially solving an optimization problem.

4.1.2 CNMM Fitting in PyTorch

While the theory behind using CNMMs as CTRNNs may have existed for quite some time, the practical use of these ideas in WBM is a recent trend. This is fueled by both an increase in computational power and the creation of software tools that make it more convenient to implement and backpropagate through custom models. One such tool is PyTorch, a relatively new Python package with built-in autograd functionality (Paszke et al., 2019). It has a tensor data structure which can track mathematical operations performed in a backwards graph. Numerical integration of differential equations can be conveniently implemented using these tensors.

In order to use BPTT with PyTorch, an error value must be calculated using an objective function which takes the simulated time series as input. This error value is used as the starting point of the chain rule used by BPTT, which continues through operations tracked in the backwards graph (that was generated during a forward pass simulation), ultimately to the parameter values being fit. This objective function could be something as simple as the last value or mean value of a state variable.

A benefit of doing gradient descent via BPTT over other types of optimization techniques such as brute force techniques or estimating the gradient by statistical sampling, is that it allows for many more parameters to be fit simultaneously. However, there are also several caveats to this approach. As mentioned previously, RNNs are known to have a problem with exploding or vanishing gradients for large sequences and the PyTorch backwards graph can use a lot of computer memory. A novel solution will be proposed for these problems in Section 5.2.2. For this gradient descent to work well, the problem should be convex such that the optimizer can follow the gradient to find the global minimum error. This will be evaluated further in Chapter 5.

The WBMG implements its deep learning methodology in PyTorch, and similar approaches utilizing PyTorch have been looked at by multiple authors, particularly in the neuroscience field (Chen et al., 2018; Poli et al., 2021; Kidger 2022).

4.1.3 Previous Implementations by the WBMG

The development for WhoBPyT expands from two studies (Griffiths & Wang et al., 2022; Momi et al., 2023). In these past works, CNMMs with some modifications were implemented in PyTorch, and the Adam Optimizer (Kingma et al., 2014) was used. Additionally, objective functions with extra components inspired by variational Bayes with prior means and variances of model parameters were included (David et al., 2006). These models used the negative Laplacian for network connections, and in addition to fitting global NMM parameters may also fit SC or LF weight gains.

The first study showed success at reproducing FC in the pseudo-time scale BOLD. A modified RWWEI was used to fit pseudo-time scale BOLD simulation to empirical BOLD FC, and to do so the time series was segmented into windows (referred to as batches in the first version of the paper). The parameters fit included some global NMM parameters, the connectivity gains, as well as some other parameters related to second level statistics (see Table 4.2) included as part of the objective function. The model was trained on a pseudo-time scale BOLD because with time constants greatly sped up, the number of times steps and thus the memory required to generate an FC matrix in the objective function was reduced.

The second study showed success in fitting time series to task evoked EEG. A modified Jansen-Rit model was used to fit evoked potential EEG. In this case, the phase could be aligned so objective function which correlated the time series directly was used in combination with some other objective function components. The parameters fit included some global CNMM parameters, the connectivity and LF matrices, as well as some extra parameters in the objective function. After fitting, the model was used to study the propagation TMS stimulation when lesions occur at various points in time.

4.1.4 Rationale for Differentiating from Previous WBMG Studies

In previous work by the WBMG, certain tradeoffs were made when choosing to incorporate various features and relaxing certain biological constraints. In this thesis, with the objective of evaluating the CNMM as a CTRNN methodology, we use a more restricted set of options. This includes choices relating to objective function components, which parameters to fit, as well as modifications to the CNMM itself.

In the previous WBMG work, additional objective function components inspired by variational Bayes were integrated as part of custom objective functions. These ideas may improve CNMM performance, but they can make it much more challenging to evaluate the performance of the CTRNN parameter fitting technique. This is due to the strong influence the initial values of the priors have on the fitted parameter values, which makes the desired gradient for an optimized parameter to be unclear. Additionally, these components are separate from the CNMM implementation and do not use BPTT, which makes it difficult to determine the CTRNNs contribution to the parameter trajectories. Thus, for the stated reasons, these objective function components are excluded from the evaluations performed in this thesis.

In the early successes of the WhoBPyT approach, some global parameters, along with the SC and LF matrices, were fit. Fitting the SC weights may be one of the most natural choices of the method as it is related to RNNs. However, SC weights are informed by neuroimaging measurements, and we instead wish to work towards fitting local NMM parameters. This is partly because, despite the presented analysis focusing on fitting to individual subjects, we wish to keep the option open to fit parameters that generalize across subjects. This would likely involve the SC and LF matrices being considered the input for each subject (as opposed to just the noise and/or stimulus), in which case they could not also be a matrix of fitted parameters. Additionally, it may only be appropriate to fit SC and/or LF weights for certain research questions. Furthermore, the LF is an additional modality after the CTRNN part, which is similar to adding a linear layer to a model, so could similarly convolute the evaluation of the CNMM as a CTRNN methodology.

With regards to modifications to the model, we keep very closely to the published RWWEI equations (see Table 2.1; Deco et al., 2014; Wong & Wang, 2006). In previous work, a negative Laplacian feature for the SC matrix was used which may bring more stability to the network and to training, however it adds a self-inhibition connection to the excitatory population, which is biologically questionable and is a model modification we do not use here. Additionally, boundary conditions on state variables may improve numerical stability, but they are avoided as they may influence the dynamics. Due to the large memory requirement of running BOLD simulations, pseudo-time scale BOLD was used in past research. One option is to use a PyTorch implementation of the model for training with faster BOLD dynamics, and then verify the found parameters on a different implementation with the original BOLD dynamics. However, in this case the CTRNN being trained on is deviating from the CNMM, which we wish to avoid in this thesis. Furthermore, the signal used for EEG may be the same or very similar to pseudo-time scale BOLD, so it may not be applicable to use this approach when fitting parameters of concurrent EEG and fMRI CNMMs.

Despite the mentioned constraints, there are still many configurations that need to be tested in order to determine how well this approach works for fitting parameters of different CNMM models, parameters, and objective functions. Developing a codebase that can conveniently run multiple experimental variations is thus quite valuable.

4.2 The WhoBPyT Python Package

WhoBPyT is a Python package in development for fitting parameters of CNMMs. It generalizes code that has been used in prior research by the WBMG. The package is being openly developed on GitHub and this thesis is based on git version ef921c3 (https://github.com/GriffithsLab/whobpyt/tree/ef921c3) of the code, from November 2023. Up to this commit, Andrew-Clappison had contributed 53 commits (excluding merge commits) for a total of 11,688 additions and 5,590 deletions according to GitHub Insights. The code has also been saved as a branch named v0.2.0 (https://github.com/Andrew-Clappison/whobpyt/tree/v0.2.0). It is still in an alpha state, so the package may change substantially in the future including the API usage and naming conventions.

In this chapter, several areas of contributions are highlighted. Firstly, I led the development to convert two prototypes from the WBMG into the format of an open-source Python package, as described in Section 4.2.1. Secondly, I worked on a second implementation of the RWWEI model (with class named RWWEI2 in WhoBPyT) that is compatible with new training features, such as batching and GPU support, as is demonstrated in Section 4.3. Thirdly, I developed classes with features to support fitting to concurrent resting-state EEG and fMRI fitting on the new RWWEI2 implementation (which are introduced in Table 4.2 and Table 4.3). The multimodal aspect will be evaluated further in Chapter 5.

4.2.1 WhoBPyT Code Architecture and API

The core of WhoBPyT is based around three class types: CNMM models, objective function components, and training paradigms which can be used in various combinations. O*bject-oriented programming* (OOP) design was followed more strictly to make it easier for others to contribute and expand the package with additional features and options. As part of this, abstract classes were defined, which provide some free functionality but also are a template as to which methods should be implemented for the new class to be compatible with the features of the rest of the package. These core classes will be further described in the subsequent sections and are illustrated in the *Unified Modeling Language* (UML) diagram of Figure 4.1. Additionally, there are some helper classes and evaluation classes that were introduced or updated to encapsulate better some features used in the original papers.

The par class is a new class created to encapsulate some of the functionality to constrain parameter values to more biologically plausible ranges and for objective function components related to the prior. One feature is that the parameter values can be constrained by storing the value of the parameter as the log of the value, and every time it is used in the equations the exponent can be applied to the stored value. In this way, the parameter value is not able to change sign. To allow a parameter to change whilst still penalizing it for moving away from the biological priors, a penalty based on the deviation from prior mean and variance is used.

The Recording class was created as an attempt to bring some standardization to transmission of data time series data between different classes. This is used as the input and output from the fitting classes, and as input into the visualization functions. This way the information is stored in a consistent way that classes can rely on, but methods are provided to convert the data into different formats, such as in a windowed format. However, this is not currently used to transmit information between models and objective function classes, which instead use a dictionary of state variable time series (dimensions of time by number of nodes, possibly with additional dimensions for various features).

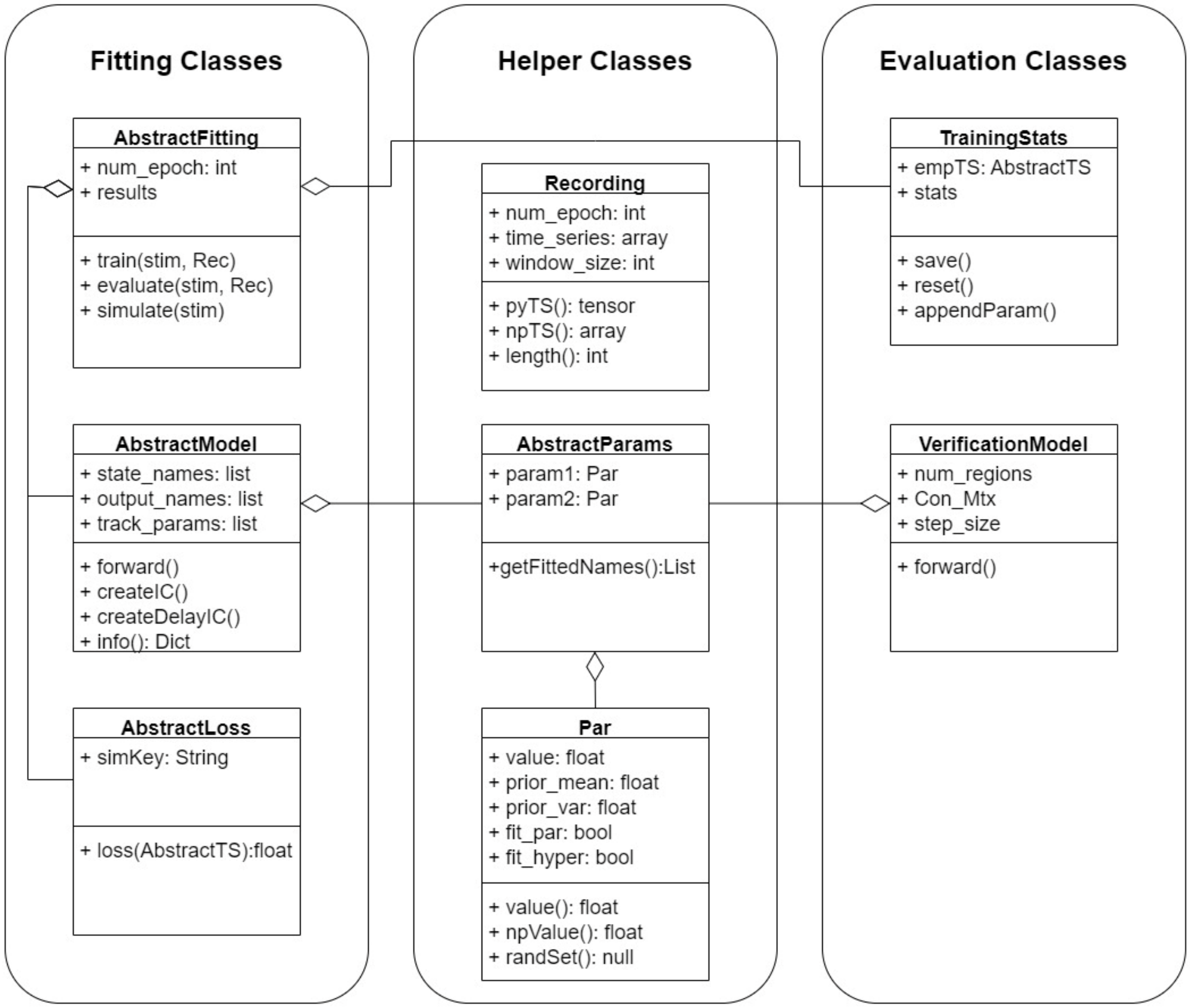

Figure 4.1 – *UML diagram of core abstract and helper WhoBPyT classes.* WhoBPyT is built around three main class types, the model classes, objective function component classes, and fitting paradigm classes. The model parameters are designed to work for both a PyTorch version for training and NumPy version for validation. The TrainingStats class is there to record information to evaluate the training performance. (Created with draw.io)

Another objective was to increase transparency through the API by having method arguments to turn on or off various techniques to improve numerical stability, reduce computational requirements, or improve optimization performance. Furthermore, the AbstractParams Class was designed to work in both the PyTorch model implementation as well as a NumPy implementation, to encourage and make it more convenient to check the parameters in a verification implementation, as described in Section 4.2.5.

The Training Stats class tracks the objective function loss as well as the parameters marked for tracking in the model over training epochs and is based on a previous OutputNM class. The user would typically be interested in tracking all fitted parameters. For example, then the user could select the parameters with the lowest loss over training, instead of just the final parameters, though less conveniently.

4.2.2 Model Classes

Model and modal classes are provided for various CNMMs and Neuroimaging processes. An abstract class for this has been developed, which also makes it convenient to add new implementations. Additionally, as will be discussed in Section 4.2.5, there are verification implementations of the models. These are implemented in NumPy. They may still be modified from the original models if the goal is to study modified models. The Params class holding the CNMM parameters should be compatible with both implementations for easy usage.

Model/Modal Name	Description	Source
RWW (RWWEI First Implementation) With integrated BOLD	Modified Reduce Wong Wang Excitatory Inhibitory Model (Deco et al., 2014; Wong Wa) with build in BOLD Dynamics (Friston et al., 2003; Deco et al., 2013) Multiple modifications in this implementation differentiate it from the original model.	(Griffiths & Wang et al. 2022)
Jansen-Rit With integrated Channel Space EEG	Modified Jansen-Rit Model (Jansen & Rit, 1995) with built in Channel Space EEG using a Lead Field (Momi et al., 2023) With Delays	(Momi et al., 2023)
Multimodal RWWEI2 (RWWEI Second Implementation) Connected with separate BOLD and EEG components. The RWWEI2 Class can also be used individually without added modalities.	Reduced Wong Wang Excitatory Inhibitory Model (Deco et al., 2014) EEG using either Parcellated LF for Channel Space EEG as performed in (Momi et al., 2023) or simply using the synaptic gating state variable as source space EEG (by setting and Identity Matrix for LF). Bold Dynamics from (Friston et al., 2003)	(**new contribution**)

Table 4.1 – Available Models in WhoBPyT

The first implementation of the RWWEI model (class named RWW) is based on the same code used in (Griffiths et al., 2022) for fitting simulated BOLD to empirical fMRI BOLD time series. The BOLD dynamics are integrated as part of the CNMM model class itself. Modifications were made from the original models, to make the code more transparent, and many arguments are added to the model's methods to indicate the modifications and make them optional. Some examples would be using negative Laplacian for the connections instead of the SC matrix itself, or adding boundaries to the state variables which modify the dynamics.

The Jansen-Rit Model is based on the same code used in (Momi et al., 2023) for fitting evoked potential EEG in channel space. The LF matrix that converts source space activity to channel space EEG is integrated as part of the CNMM model itself. As done in the previously mentioned model, modifications were made to this model and to make the code more transparent, many of which were added as parameters to the model classes methods.

The Multimodal RWWEI model is another implementation of RWWEI which follows the (Deco et al., 2014) equations more closely, with fewer modifications to the original dynamics. Furthermore, it is a subclass of a RWWEI2 class with extra modalities of EEG and BOLD added. If one desires to use source space EEG instead of channel space EEG the LF matrix can simply be set to be the Identity matrix. This model has some newer features, such as support for the blocking and batching technique combined with the ability to fit local (per-node) values for certain parameters. In order for local parameters to be fit, some modifications to the implemented equations may need to be done, and the par object should hold a vector the size of the number of nodes instead of a scalar. GPU support was added to this class.

Other models can be created by extending the abstract model class. It is easiest and recommended to add the modalities as part of the model instead of using the modality subclasses.

4.2.3 Objective Function Classes

There is an abstract class for objective function components, with multiple types integrated into WhoBPyT. Objective functions components measure some phenomena in the simulated time series and compare to an empirical target. The empirical data may be in the form of a time series for which the objective function calculates the phenomena the same as in the empirical data, or it may be in the format of the abstract phenomena such as a FC matrix or Power Spectral Density already. Also, the difference between the simulated and empirical target is usually in the form of a correlation or *mean squared error* (MSE). The choice of the objective function will depend on the research question and the available data. Custom classes can be made which combine different standards or model specific components.

Objective Function Component Name	Description	Source
Functional Connectivity Correlation	Correlation between lower (or upper) triangle (LT) of empirical and simulated functional connectivity. Phase Invariant for resting-state data. There is a windowed version. There is a fixed target FC version with GPU Support (**new contribution**). $-\log(0.5 + 0.5 * Cor(LT(SimFC), LT(EmpFC)))$	(Griffiths & Wang et al. 2022)
Time Series Correlation	Direct correlation of times series for task-based neuroimaging data. $Cor(SimData, EmpData)$	(Momi et al., 2023)
Second Level Statistics	Loss based on prior mean and variance of a parameter. These may be included in custom objective functions and can be implemented differently than the example below. $\frac{(parameter\_value - prior\_mean)^2}{prior\_variance}$	(Griffiths & Wang et al. 2022)
Target Mean State Values	Difference between target value and local mean value of a state variable over time. With GPU Support. $MSE(\frac{1}{T}\sum_{t=1}^{T} V_{n,t} - TargetValues)$	(**new contribution**)
Power Spectral Density	Phase Invariant comparison of oscillation powers for resting-state data. Mean Squared Error of Fixed Target PSD. With GPU Support. $MSE(FFT(SimData)^2, FFT(EmpData)^2)$	(**new contribution**)

Table 4.2 – Available Objective Function Components in WhoBPyT

4.2.4 Fitting Paradigm Classes

Parameter fitting classes perform the training of the parameters using various optimization or machine learning paradigms. The focus so far has been on optimization paradigms, but machine learning paradigms are possible too. When training, it is useful to keep track of model performance over training windows/epochs. Also, to keep track of the fitted parameters over training epochs. This is done by a sub class training stats that the fitting class has an object of.

Fitting Paradigm Name	Description	Source
Model_fitting	The original fitting paradigm of the method. Uses windowing. Can also just have 1 window to be regular fitting.	(Griffiths & Wang et al. 2022)
Fitting_Batch	Batched training, running multiple independent simulations simultaneously for one parameter update. Expected to have a speed boost when working with GPU.	(**new contribution**)
Fitting_FNGFPG	Specialized for the case where longer simulations are required. This is used for simulating true-time scale BOLD.	(**new contribution**)

Table 4.3 – Available Model Fitting Paradigms in WhoBPyT

Model_fitting is the original method used in (Griffiths et al., 2022). It uses a windowed approach, where segments of a larger time series are fit to through individual backpropagations. Previously this may have been referred to as batching. Windowing can be interpreted as a special case of Truncated Backpropagation Through Time. That is, instead of backpropagating through the time series once at the end of the simulation, only a recent number of iterations are backpropagated though. If the window size is the entire time series, then this would be a regular fitting.

Fitting_Batch is a batched version of fitting, particularly created to attempt to maximally exploit GPU capabilities to perform operations in parallel. Multiple empirical timeseries are fit to at the same time for each update of the parameters. If the batch size is one, then this would be a regular fitting.

Fitting_FNGFPG stands for Forward No Gradient Forward Parallel Gradient, and was designed to run true time scale BOLD at reduced computer memory requirement. This is described in further detail in Section 5.2.2.

Training can be sped up by running the training on GPUs. GPU support was added to all three of these paradigms. Batch_Fitting benefits the most from this, though the others can still benefit as well. For this to work, the model and objective function classes must also have GPU support.

4.2.5 Practicalities for using WhoBPyT in CNMM Studies

When performing parameter fitting with WhoBPyT, there are many options available to the user for which the preferred configuration would largely depend on the research question and the available data. There are multiple built-in models which can be used with various objective function components and parameter fitting paradigms. Within each of these there are various features, options, and modifications that should be considered when performing parameter fitting and evaluating the results.

Typically, RWWEI has been used for BOLD studies and Jansen-Rit has been used for EEG studies. A particularly important choice is which parameters to fit. Essentially all model parameters can be readily fit on a global level, and some model parameters can also be fit on a per-node level. The connectivity between nodes, either considered SC or effective connectivity, would be the values regularly fit in the context of RNNs. However, from the perspective of WBM, perhaps fitting the global or local parameters of NMM nodes themselves would more commonly be useful for research questions.

The implementation of these NMM models and modalities in PyTorch may have substantial built-in or optional features and modifications. These could be for purposes of numerical stability, optimization algorithm performance, model performance, or due to computational resource constraints. Examples include using the negative Laplacian of the SC for the connectivity matrix, optional delays, using faster dynamics, using boundaries on state variables, and using bounds on fitted parameters. Thus, it is important to look at the arguments of the model class methods as well as to verify the found parameters using NumPy verification implementations or other software. The verification versions are the original model likely with fewer modifications for optimization performance or numerical stability. As the code is in development, it is strongly recommended to either review the verification implementation directly or to use more mature software such as TVB for verification.

Another factor, which may have influenced which CNMM was chosen, is which parameters to fit. There may be known biologically relevant ranges for parameters, or certain interpretations of the parameters such that it is invalid for the parameter to change sign. Furthermore, depending on the parameter, large changes from the original value could cause numerical instability. For example, changes in time constants could change the appropriate step size of the numerical integration. Thus, features of having prior mean and variance for the parameters influence the gradient and/or storing the log value of the parameter can be used, so that the parameter value itself never goes negative. That said, it is recommended to first try to train the model without these boundaries as they can affect the gradient descent, and ideally the model will naturally tend towards biologically relevant parameters without these constraints. Whether this happens in practice will depend on the model and the dataset used.

Which objective function to use is the next natural decision. The objective function should be able to find a parameter range that optimizes phenomena in the CNMM model relevant to the study. This could be one objective function component or a custom objective function class with multiple components.

The paradigm chosen may depend on factors such as the data available, the computational constraints, and what kind of generalizing ability one wants the model to have. Hyperparameters of the fitting paradigms can include the learning rate, window size, batch size, block size, and objective function components. Model parameters could also be hyperparameters if not being fit by the optimization algorithm.

What is considered the input and output of training may depend on the research question and will be a factor in whether to interpret the parameter fitting as an optimization or machine learning paradigm. That is, whether the entire dataset should be used for training, or whether the data should be split into training, validation, and testing sets. This will usually be related to what the found parameters are to represent. For example, if the researcher is interested in optimizing the parameter of the model to reproduce the fMRI output of a particular individual, and possibly compare parameters across individuals, then this would be an optimization problem. On the other hand, the researcher may want to, based on some measurement from an individual such as a SC matrix, have a model that can predict the FC. In this case, the input would be SC and the label the FC. An example research goal could be to have a model which generalizes to work for individuals for which corresponding fMRI FC is not available, which would be a machine learning problem.

Many NMMs use noise, so in a sense every training run uses unique input, even in the case of resting-state data. This could be considered equivalent to having a training dataset with infinite data augmentation. Since data augmentation of a training set element should not be put into validation or testing, one could argue that noise should not be interpreted to change the optimization problem to a machine learning problem in general.

4.3 Demonstration using the New RWWEI Implementation

The demonstration presented here is to showcase the new implementation of the RWWEI model (class name RWWEI2 in Table 4.1), with some associated capabilities and features added to WhoBPyT. This includes fitting local NMM parameters, as well as a batching paradigm and GPU support. Furthermore, to verify that a simple parameter optimization is working as expected.

The RWWEI Connectome-based model (Deco et al., 2014) uses the same base model that was analyzed in Chapter 3, but with multiple NMM nodes connected in a network with additional neuroimaging measurement modality dynamics added. In this demonstration, we used a synthetic SC matrix for the connection weights, which is illustrated in Figure 4.2.

Something new to WhoBPyT work was fitting local NMM parameters. Here the local J_i values were fit to perform FIC. The FIC method, an algorithm associated with the RWWEI model, adjusts the local nodes' J_i parameters (inhibitory population to excitatory population strength) to ensure the mean firing rate of each node in the network is near 3Hz, corresponding to an S^E value around 0.164. These values will be dependent on the SC used in the model. This is presented in Deco et al. 2014 using an iterative algorithm. However, it should be a simple task for WhoBPyT to do this using backpropagation and it does not require empirical data.

The new Target Mean State Values objective function (see Table 4.2) was created specifically to perform a version of FIC for RWWEI, though other uses are possible. This works by simply calculating the MSE between the average value of the state variables for each node and the target value.

The demonstration used the Fitting_Batch (see Table 4.3) optimization paradigm. This was chosen as it is a simple objective function which only needs to run for a short simulation time, so there is more memory available for the batch dimension. This can lead to a smoother gradient for backpropagation to follow.

Analysis was performed on the Digital Research Alliance of Canada – Advanced Research Computing resources (alliancecan.ca). The model, objective function, and fitting paradigm used all had GPU support, so either CPU or GPU could be used. Running time of CPU took approximately 1035 minutes, while running time for GPU took approximately 35 minutes, roughly a 30x speedup in this case. Figures generated in Python.

We see that WhoBPyT successfully performs FIC, as seen in Figure 4.2. The loss rapidly decreased and then stabilized near zero. Similarly, the fitted J_i parameters rapidly increased to bring down the mean firing rates and then stabilized with a slower drift. This was the desired behavior of the optimization approach.

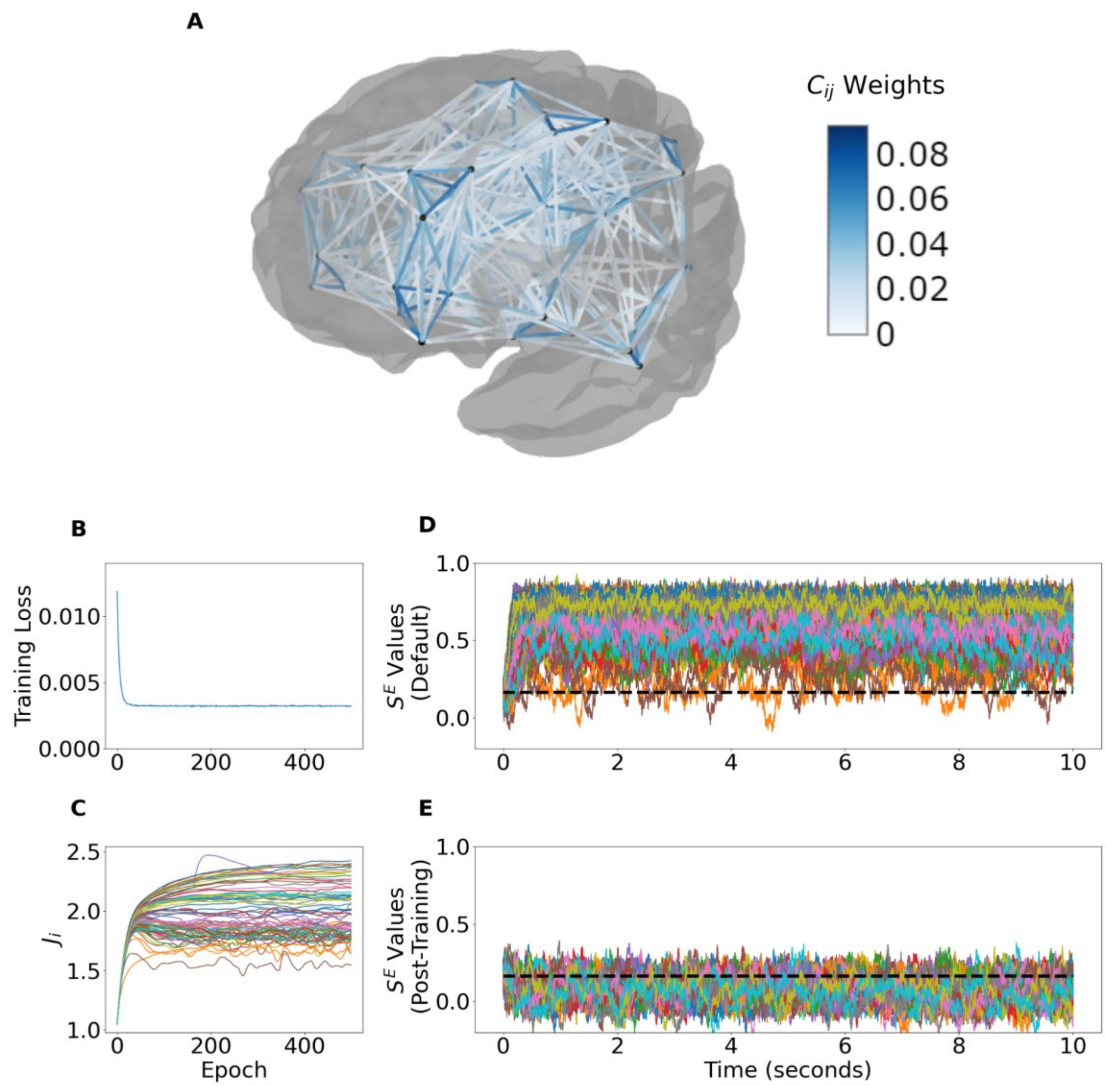

Figure 4.2 – *WhoBPyT used to fit local J_i parameters with the mean S^E objective function (performing FIC). Panel* A: Visualization of the synthetic SC weights used (Created with Nilearn). **Panels B&C:** Training Loss and J_i parameters changing over training epochs. **Panels D&E:** Simulation of S^E values with original and FIC fit J_i parameter values. We see that the error rapidly decreases, and the J_i parameters stabilize, to achieve mean firing rates closer to the target (dotted line).

Training Specifications	
CNMM Model	RWWEI2
Objective Function	Average Local mean S^E Value
Optimization Paradigm	Fitting Batch
Parameters fit	Local J_i's for all 80 nodes
Number of epochs	500
State Value Initialization	Reset after each backpropagation
Simulation time	0.1 seconds by 300 batch
Optimizer	Adam
Learning Rate	0.05
Device	GPU

Table 4.4 – Training specification for fitting local J_i parameters to perform FIC.

4.4 Discussion

4.4.1 Summary

In this chapter, the WhoBPyT CTRNN methodology has been positioned within the existing literature. That is, CNMMs as custom CTRNNs where the nodes are based on NMM equations and using BPTT for parameter fitting. The WhoBPyT package architecture was defined, and new classes and features were integrated into the framework. A demonstration of a subset of these new features was performed by doing a gradient descent via backpropagation version of FIC for the RWWEI model.

This project has contributed to progressing the codebase towards OOP design with abstract classes and improved encapsulation. This was required to make it easier for developers to integrate new code into the package now and in the future. Additionally, the various features of the methods were made more transparent by adding them as optional method arguments. This makes it easier for users to choose and document what features were included in their analysis.

This chapter revolved around the RWWEI Version 2 implementation. This implementation focused on being a multimodal model that is also more consistent with the original RWWEI equations (see Table 2.1; Deco et al., 2014; Wong & Wang, 2006), with some minimum modification to the firing rate equation for numerical stability. Fitting FIC was the first attempt of fitting local node parameters. It worked quite well, as can be seen in Figure 4.2.

GPU support was added to part of the package for this thesis work. This includes the RWWEI2 model, with multiple objective function components and fitting paradigms. It resulted in large speedups, more so in some cases than others. Unlike working with pre-implemented model components in PyTorch or TensorFlow, features such as batching or GPU support had to be manually implemented in each class being used together to support such functionality. In the demonstration of this chapter, the GPU support showed an order of magnitude increase in speed.

So far, this WhoBPyT CTRNN technique with BPTT has focused on fitting models to individual subjects. This is different than regular ML, and can instead be considered optimization. That is, all fitted parameters have predefined physical meaning. When training in ML, it is typical to try a grid search of many different hyperparameters including modifications to the ML architecture. In our case, where the model and interpretation of the parameters is more constrained, the effort is placed elsewhere. Here we try to improve the objective function, which is only expected to succeed for convex problems.

Some tradeoffs were made to try to get the model to maximize biological relevance. Additionally, the goal was to evaluate the performance of the technique with various objective functions, as opposed to producing a parameter set which has the best performance.

4.4.2 Future Development

Further development of the WhoBPyT package will be required before it is ready for release. The API is evolving to support various features, including more supported combinations of the available models, objective functions, and fitting paradigms.

The code contributions related to this thesis have focused on support for the RWWEI Version 2 implementation, using objective functions based on fixed targets. Additional features already available in other classes could include using delays and support for fitting to time series based empirical data. Since we are using RWWEI for generating EEG it would make sense that we would want the option of delays. However, they may not work well with some fitting paradigm approaches. In the other direction, the RWWEI Version 2 has some features not yet integrated into other classes. GPU support and Batching have yet to be integrated into the other model implementations or time series-based fitting paradigms and objective functions. Different levels of feature support at this time also leaves some barriers on full support for the interchangeability of classes.

From the API perspective, improvements could be made to simplify the user's involvement with preparing data with particular time lengths and sampling rates, as well as resampling of output data. It may make sense to have resampling performed between classes as well, such as when calculating FC matrices. It would also be good to give users easier control over the initial conditions of state variables used during training. Perhaps, a feature of using fixed points for initial conditions from Chapter 3 could also be integrated. Finally, features for convenient plotting could also be provided, which could include linear stability analysis figures similar to the analysis performed in Chapter 3.

Additional models, objective function components, and fitting paradigms could be supported. For example, a linear model was briefly studied but not fully integrated, and other models are being integrated into the package by other lab members. Objective functions based on linear stability analysis as described in Chapter 3 could be interesting (that operates directly on the equations as opposed to the time series), particularly for promoting parameter sets for which the model oscillates. That is, extending WhoBPyT to some other backpropagation-based gradient descent techniques which do not necessarily fall in the realm of RNNs. ML paradigms so far have been designed mostly to fit models to individual subjects, for which more functionality could be added to support more standard ML paradigms, where the goal is for certain parameters to generalize across subjects.

The verification models currently or to be implemented could benefit from using just in time compiled code, similar to packages like Neurolib and TVB. This can be done using the Python Package Numba (Lam et al., 2015), with some minor code changes to support it.

Finally, some thorough unit and integration tests should be created and added as part of continuous integration. This could help speed up development, as well as promote more confidence in the codebase, and similarly improving its reliability.

4.4.3 Further Evaluation

Several combinations of RWWEI2 with EEG and BOLD modalities are now supported for objective functions based on fixed targets. These were created for the goal of fitting a CNMM to concurrent EEG and fMRI datasets. These require more complex evaluations than what was done for the FIC demonstration. In addition to involving more complex phenomena than mean firing rates, there are many different combinations of parameters that can be fit (both globally and locally), for which only a subset of combinations have been tested. This leaves a lot of room for exploration.

It has been recently shown using a gradient descent technique not using backpropagation that fitting the weights of the SC, combined with the FIC method previously discussed, the RWWEI model can simulate BOLD with FC that is near perfectly correlated with empirical FC (Schirner et al., 2023). It assumed the strengths in FC are a result of connections between two regions. However, WhoBPyT is using backpropagation to find the gradient which may identify other mechanisms. For example, the correlation between two regions may not be due to direct connection between them, but instead they are a result of a third population inputting into both. So, in a way similar to how WhoBPyT was used as an alternative implementation for FIC, it may be possible to provide an alternative approach for this newer paper.

In the demonstration of this chapter, the task gradient and the backpropagation gradient appear to be well-aligned for balancing the firing rates of the network to 3Hz. This may not be the case in general, and there are many different objective functions to consider. Dealing with multimodal problems introduces more challenges with additional objective function components that can be further combined in different ways. This is what Chapter 5 will address next, testing whether backpropagation finds good gradients in the individual components, and then whether a benefit can be achieved by fitting multiple components together.

CHAPTER 5 – DEEP OPTIMIZATION: MULTIMODAL RESTING-STATE CNMMS

Abstract

Here we work towards the goal of extending the CNMM CTRNN technique to fit parameters of a Reduced Wong-Wang Excitatory Inhibitory (RWWEI) CNMM model (see Table 2.1; Deco et al., 2014; Wong & Wang, 2006) to concurrent resting-state EEG and fMRI. The task is optimizing model parameters to real or synthetic EEG and/or fMRI data.

Fitting to concurrently recorded resting-state EEG and fMRI requires us to extend previous single-modality work in two ways. First, we need to create and evaluate a time-invariant objective function for resting-state EEG. Second, by creating a fitting paradigm where running true-time scale BOLD becomes computationally feasible on the timescale required for calculating Functional Connectivity (FC) matrices. Synthetic data where the ground-truth gradient and local optimum is known is used to simplify the evaluation of these ideas. They are both evaluated individually first, unfortunately showing only moderate success, before moving on to multimodal objectives.

Given the existing challenges with resting-state EEG and true-time scale BOLD parameter fitting among other factors, the multimodal problem considered here has been simplified in several ways. In the first multimodal evaluation, the model starts with parameters for which it oscillates using the brute force technique described in Chapter 3, then uses the deep learning technique to try to maintain those oscillations while attempting to improve the fit of the model to empirical FC. In a second multimodal evaluation, an experiment is done training the model on source EEG FC, fMRI FC, and on combined source EEG and fMRI FC, to compare their respective performance.

The chapter ends with a summary of what has and has not shown success so far in these investigations, and suggests potential reasons why. Some ideas are discussed of what could be added to get closer to the goal of multimodal parameter fitting. Ultimately, this goal will have been achieved if a fitted model can reproduce the characteristics of both modalities individually as well as showing the phase relationship between the EEG and fMRI modalities.

5.1 Introduction

The goal of this chapter is about evaluating the training of the RWWEI2 implementation in WhoBPyT with various objective functions. The evaluations are all related to the goal of fitting the model to concurrent resting-state EEG and BOLD data.

5.1.1 Multimodal CNMM Parameter Fitting

Multimodal CNMMs generate concurrent time series for multiple modalities, representing different physical aspects across different time scales for the same underlying system. For example, EEG which has high temporal resolution with BOLD which has high spatial resolution (Ritter & Villringer, 2006). When fit to empirical data, CNMMs theoretically have the potential to be more accurate and biologically meaningful than creating a separate model for each modality. Parameters can be further constrained when being fit to multiple modalities simultaneously. Furthermore, the accuracy of each modality could be individually improved compared to focusing on one modality.

When fitting multimodal CNMMs, objective function components are needed for each modality, and these may be in competition with each other. In the case of resting-state EEG, it is often the frequency components we are interested in which are grouped into various frequency bands. In the case of resting-state fMRI BOLD, it is often the FC matrix which relates the correlations between all combinations of brain regions.

When working with multiple modalities it may become more important to represent them in their physically accurate time scales than when studying them separately. This is computationally challenging, because representing the true-time scale of BOLD can require millions of CNMM time steps. Not only is this challenging because of the associated run time and memory requirement, but also because RNNs of this size are known to have vanishing or exploding gradient problems.

Multimodal models have additional potential to study the relationship between modalities, compared to their single modality counterparts. For example, the ability to capture the phase relationship between EEG and fMRI, to study the EEG alpha power vs. BOLD anticorrelation. This is because, even though the phase of the model cannot be aligned with empirical resting-state data, the phase across the modalities of EEG and fMRI is aligned.

5.1.2 Concurrent Resting-State EEG and fMRI Data

There are many public neuroimaging datasets, but few of these are concurrently recorded multimodal datasets. Concurrent resting-state EEG and fMRI studies tend to be smaller in sample size, often with fewer than 20 individuals. This may in part be due to the specialized equipment required to capture concurrent EEG and fMRI. Deligianni & Clayden provide one such dataset that includes fMRI in voxels, EEG at the channels, and associated SC for the subjects (Deligianni et al., 2014; Deligianni et al., 2016).

Multimodal datasets require specialized processing pipelines for which choices may depend on the research question (Huster et al., 2012). When neuroimaging data is collected, many technical artifacts can enter the data which are of non-biological origin and thus, the data often must go through data processing pipelines. This includes some extra processing steps due to modalities interfering with each other. For example, the fMRI pulses must be removed from the EEG signal.

Before analyzing the data, it may be necessary to align the modalities spatially and temporally. When EEG is being studied with fMRI it can be analyzed directly in the channel space, but often it is instead studied after being transformed from channel space to source space. Then this space is parcellated into regions and the mean activity of each region is calculated. To temporally align the concurrent data, fMRI scan time points recorded in the EEG markers can be used.

In the final multimodal evaluation (Section 5.3.4), available SC and precision matrices (inverse of FC matrices) for EEG and fMRI are used, obtained from the supplementary information (Deligianni et al., 2016). These are based on concurrent resting-state EEG and fMRI recordings with SC matrices also available. This is useful as it is a public dataset already in a processed form for the analysis to be easily replicable. Additionally, a raw form of the data is publicly available online, for which custom processing can be used, as required, for specialized research questions.

5.1.3 Synthetic Data Generation

Synthetic data is convenient to use for evaluating the performance of optimization algorithms, particularly when phenomena of interest can be explicitly integrated into the data so that there is a known ground truth that the model can be compared to. In this chapter, a synthetic SC matrix is used, generated by a function that is integrated as part of WhoBPyT. This function works by sampling nodes in 3D space (with some constraints) for one hemisphere, then mirroring the node locations to the other hemisphere, so that the network is symmetric. Subsequently, the connectivity matrix weights are calculated based on the distance to other nodes. A challenge with this kind of synthetic SC is that a corresponding LF matrix is unknown, hence EEG must be analyzed in source space.

With a synthetic SC, a CNMM can then be run to generate synthetic EEG and fMRI time series. Target PSD or FC matrices can then be calculated, for which we know the model can reproduce very accurately. This can simplify the problem of evaluating the learning algorithm, as in the case of fitting to empirical neuroimaging data, it may be unclear how well the model can reproduce the data with optimal parameters.

5.2 Methods

5.2.1 Fitting Power Spectra

An objective function was needed to be able to fit to resting-state EEG data. It is unclear how one would align the phase of the simulated and empirical resting-state time series. So instead, the focus is on the oscillation frequencies present in the signal. A common way to look at frequency components in a signal is with a Power Spectral Density (PSD). Thus, a PSD based objective function is proposed. For this to work, consistent processing on both the simulated and empirical data is required, as well as averaging of PSDs to make them smoother.

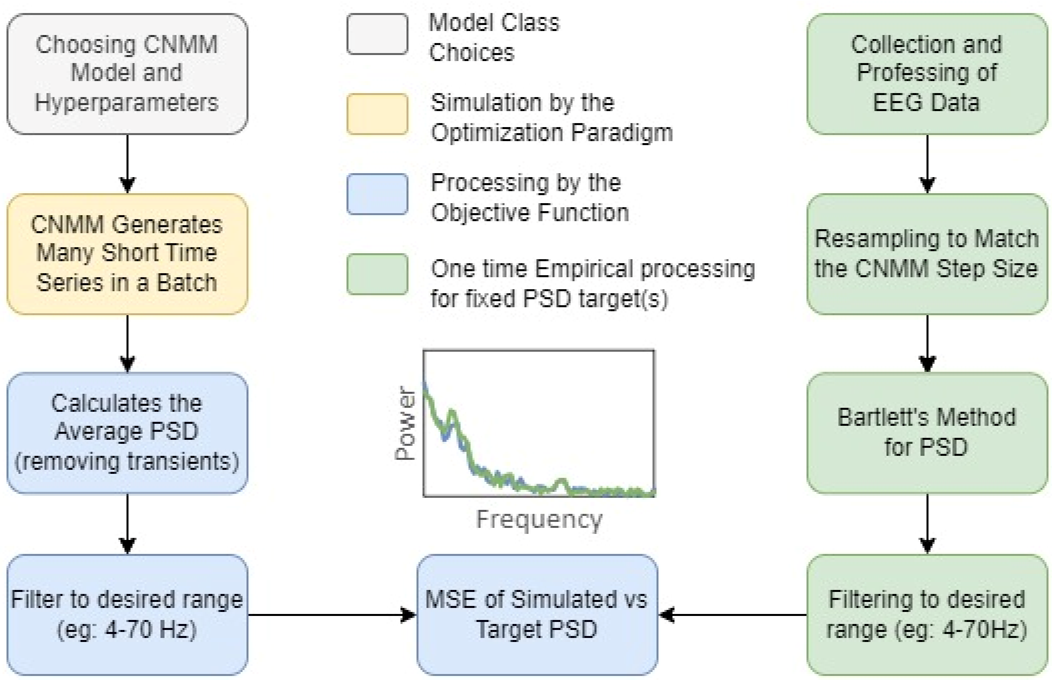

Figure 5.1 – ***The PSD objective function for fitting to resting-state neuroimaging data, particularly EEG.*** Simulated data is generated and processed in a consistent way with the target data. The loss is the mean squared error between the simulated and target PSD. (Created with Excel and draw.io)

The EEG PSD objective function calculates the loss as the MSE between the simulated and target PSD. This being the case, it is important that the magnitudes of the simulated and target PSDs are similar. This might require resampling to the same rate, removing the mean from the signal, and extracting a frequency range of interest. Various other processing steps may also need to be considered and made consistent between the simulated time series and empirical data when being compared in the objective function. This includes removing the transient by having good initial conditions or removing an initial section of the time series. The learning rate and whether state variables are reset upon each backpropagation step are additional factors. The approach implemented in WhoBPyT and used in subsequent analysis is shown in Figure 5.1.

In empirical and simulated EEG data the time step can be quite small on the order of 0.0001 seconds which results in PSDs measuring up to 5000 Hz when only 1 to 70 Hz may be relevant for typical EEG studies. Additionally, simulation lengths may be in the range of seconds to minutes. This signal from empirical neuroimaging data can be very noisy, and multiple methods for smoothing exist. These included downsampling, Bartlett's method, and Welch's method. Bartlett's method is chosen for the present study, due to the smoothed result, while being simpler to implement in PyTorch than Welch's method. Bartlett's method is implemented simply by dividing the signal into equally sized non-overlapping bins, calculating the PSD for each of those bins, and returning the average of the windowed PSDs. The EEG PSD objective function can be used on times series of short 1 second lengths, so that the PSD density is one point per Hz on the x-axis. To better utilize computational power of the GPU, many segments are run in parallel as a batch and averaged together in the simulated PSD case. In this way, it is similar to using Bartlett's method, except that it uses independent sections of simulation as opposed to adjacent sections of one time series. In this sense, it is not regular batching as there is only one error value. This means that regular batching can be performed in addition to this.

5.2.2 True-Time Scale BOLD

In previous work done by the WBMG (Griffiths et al., 2022), faster fMRI BOLD dynamics were used. This was done by treating the output of the NMM in milliseconds as seconds into the BOLD dynamics, which is in effect changing the time scale, and thus we will refer to this approach as pseudo-time scale BOLD. This was further combined with windowed segments of BOLD instead of generating the whole empirical time series worth of simulation, around 30 second windows would be done at a time. The effect of this is to greatly reduce the number of time steps required to perform backpropagation, which reduces the computer memory required and accelerates fitting speed. For the evaluation of trained model parameters using this approach, it becomes particularly important that the performance of the fitted parameters be verified on another model implementation with the true-time scale BOLD.

While using pseudo-time scale BOLD can be a useful approach for training when using only a single neuroimaging modality, when working with concurrent EEG and fMRI - which have vastly different time scales - it makes sense to represent each time scale accurately. Thus, a fitting paradigm was needed to be able to fit to true-time scale resting-state fMRI BOLD signals.

The *Forward No Gradient Forward Parallel Gradient* (FNGFPG) paradigm was created to reduce the memory requirement of the PyTorch backwards graph, so that longer simulations could be backpropagated through. It works by first running an initial simulation of the full time length desired, in order to acquire the initial conditions for various segments in a subsequent forward pass. These initial conditions are dependent on the same noise and same step size being used later. During the initial forward pass, the backwards graph is not generated as that would use an impracticably large amount of memory. Then in the second phase, the different initial conditions are run in parallel, by adding another dimension to the differential equations state variable input. This time, the backwards graph is tracked while simulation goes on. When the segments are realigned, it should produce the exact same time series as that of the initial run. This is visualized and compared to other fitting paradigms in Figure 5.2.

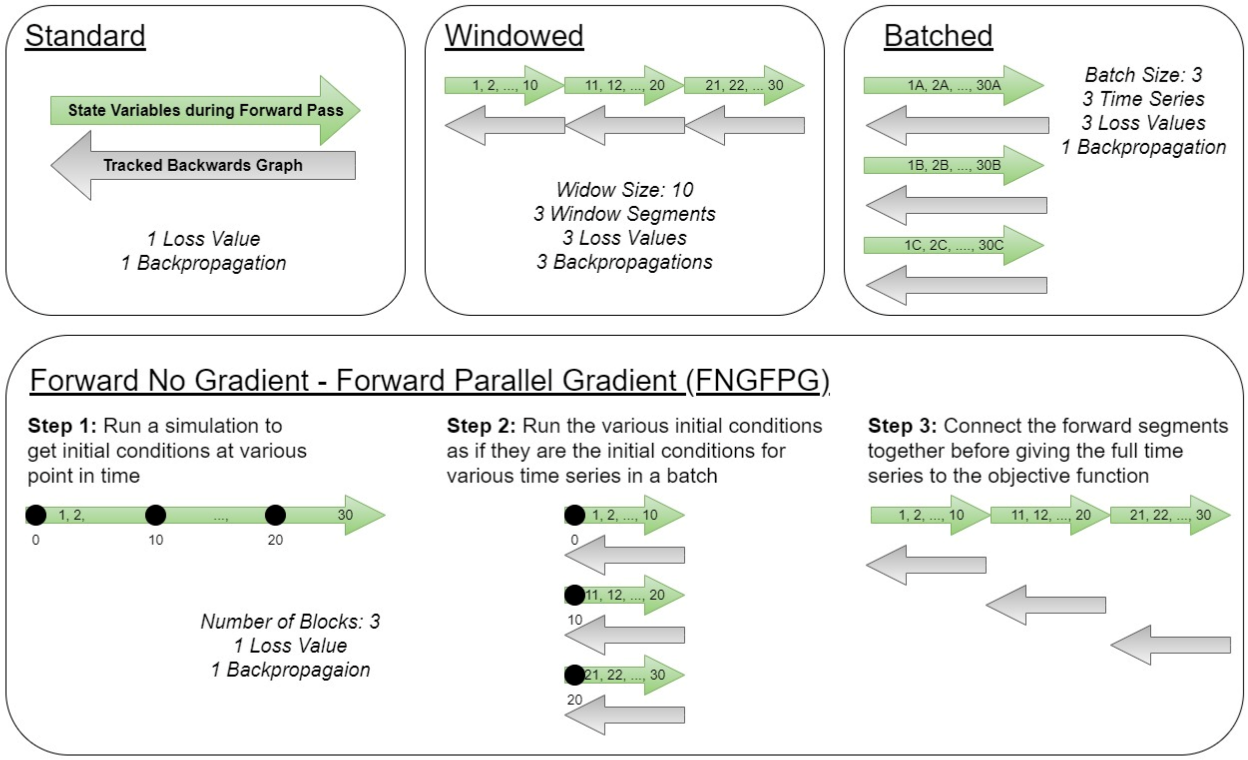

Figure 5.2 – ***Comparison of Windowed, Batched, and FNGFPG fitting paradigms implemented in WhoBPyT.*** The green arrows represent the state of the system at various time points. The grey arrows represent the backwards graph being generated by PyTorch. Windowing is useful to reduce computational requirements for dealing with longer time series. Batching is useful for shorter time series where extra memory is available for faster performance. The FNGFPG paradigm is proposed for the case of using an objective function that requires a longer length of simulation (as in the case of BOLD FC). (Created with draw.io)

The FNGFPG approach takes advantage of an understanding of the backward graph PyTorch generates to perform backpropagation, and with an understanding of stochastic differential equations simulations. Through experimentation, it was found that this backwards graph takes up a lot of computer memory - much more than the forward simulation. Furthermore, it was discovered that increasing the dimension of the forward pass did not add much memory when compared to increasing simulation time. This could be explained by a fixed cost component to create a tensor object in PyTorch and doing each iteration in the model, which could be large relative to the data stored in one iteration. Thus, if we can run a shorter simulation with fewer but larger tensors, the variable cost to fixed cost ratio increases and better utilizes the computational resources. With regards to differential equations, we do not have an analytical solution so we cannot calculate the value of a variable at some future time without simulating all the time in between. However, once we have calculated a time point, we can subsequently re-simulate any segment of time using the states of the system at one time point before as initial conditions. In the case of stochastic differential equations, this also requires using the same noise in the FPG simulation as was used in the proceeding FNG part.

With the FNGFPG paradigm, it was found to be possible to simulate and backpropagate through true-time scale BOLD of a CNMM on the order of 10's of seconds as opposed to just several seconds with 16GB RAM and a 12GB GPU. Even if unlimited RAM was available, the FNGFPG approach has shown increased speed in practice and has reduced the memory required to be able to run on GPU. The exact performance boost may depend on the autograd software the model is being implemented in and the hardware it is being run on. There may also be benefits in terms of reducing the vanishing/exploding gradient problem associated with RNNs.

Being able to backpropagate through longer simulations is very useful, as a typical way to fit a connectome-based model to resting-state fMRI BOLD is to look at the correlations between the signals of different brain areas in the format of a FC matrix which requires longer time series. Given the slow time scale of BOLD, the time series should be of the length from 10's of seconds to 10's of minutes. For whole brain simulations to match this, the Euler-Maruyama simulations with the differential equations being evaluated with timesteps of around 0.1 milliseconds, the recurrence would ideally be able to occur millions of times.

There are some caveats of using the FNGFPG approach. Firstly, there will be some breaks in the chain rule of the backwards graph, as time will not be connected between segments. However, given that RNNs experience the exploding/vanishing gradient problem for larger simulated input, this might in fact produce better results. Secondly, though the amount of memory is reduced, there is extra computation required to get the initial conditions. Luckily, the memory requirement has been reduced sufficiently that the memory required is available in current day GPU sizes, and this can improve the running time overall. Currently, the result in training performance and speed is evaluated through numerical experimentation, but perhaps further work could provide some theory to calculate an expected performance boost.

5.2.3 Multi-objective functions

A multi-objective function can be constructed as a weighted sum of multiple components. These can be used for multimodal models, or where there are additional criteria for a single modality model. The weightings are set to one by default, but can also be treated as hyperparameters to be tuned if necessary.

The general expectation and hope is that fitting to multiple modalities simultaneously will result in a model that can reproduce each individual modality better than when trained separately, due to the extra information. This is assuming the model and optimization algorithm is well defined, and the training data is of good quality. However, it is also possible that objective function components might be in conflict with each other.

Here, multi-objective functions having phase-invariant objective function components for concurrent resting-state EEG and fMRI are particularly examined. In particular, we use PSD for EEG and FC for fMRI.

5.2.4 Evaluating the Deep Learning Technique for Parameter Optimization

This thesis explores the use of a deep learning technique for optimizing the parameters of physiological models for individual synthetic or human subjects. This means that, although a deep learning technique is used, a machine learning paradigm for evaluating the generalization ability of parameters is not. The learned parameters are not (necessarily) intended to be generalizable to other subjects, although this may happen by chance. The parameters do, however, generalize to the noise the model receives as input. This noise is different for every simulation that is backpropagated through, so there is no risk of overfitting to the noise from an ML perspective.

When evaluating the optimization technique for fitted parameters for individual subjects, there is less emphasis on the absolute CNMM performance. Instead, we wish to know if the algorithm is successfully following the gradient using backpropagation. If there is a known local optimum within a parameter range, then we evaluate whether during training the parameter tends towards the local optimum from different starting points on that range. For this purpose, extra objective function components which may involve having prior means and variances for parameters are not used, as that would confound the gradient (see Section 4.1.4).

Alternatively, particularly when there are many parameters, it becomes much more challenging to identify a local optimum to evaluate gradient descent on. Thus, it becomes necessary to evaluate based on the performance of some metric(s) based on the models simulated output, as opposed evaluating the distance of the found parameters to their known local optimum values. Verification models implemented in NumPy are used when performing the evaluation. The RWWEI2 validation model is implemented in NumPy and provided as part of WhoBPyT.

There may be a concern of overfitting the model from a modelling perspective, where the model is supposed to be the correct representation of a physical system, and parameter values that produce the expected output but are not physically realistic would be considered a problem. Part of this problem is inherent to the chosen CNMM, but it can also be evaluated by how far the fitted parameters are from their original values.

Though not the focus of this thesis, machine learning paradigms involving generalizing parameters across subjects are possible. If desired, and the required data is available, this technique can be used for ML questions, where the evaluation would require splitting the data into training, validation, and testing sets.

5.3 Results

We present the results of fitting the RWWEI to individual modalities with known ground truth local optimums, then proceeding to evaluate the performance on a multimodal dataset. These experiments were designed to be easily accessible and reproducible. Thus, synthetically generated or publicly available data was used. In particular, the already processed SC, EEG FC, and BOLD FC matrices publicly available as a supplementary appendix (Deligianni et al., 2016). The same WhoBPyT version was used as described Section 4.2. Training was done using a PyTorch implementation, and the final evaluations used a NumPy validation version, as explained in Section 4.2.5.

The training for the following analysis was run using the Digital Research Alliance of Canada - Advanced Research Computing resources (alliancecan.ca), either on CPU or GPU (NVIDIA P100 12GB) on the Graham cluster. The jobs were submitted as Slurm array jobs. Evaluations were done using 5 minutes of simulation on a verification implementation of the model, which required 16GB of memory. The simulations were first down-sampled before calculating the FC matrices that are compared to empirical data. Figures were generated in Python unless otherwise specified.

5.3.1 Fitting Power Spectra of Resting-State EEG

The Power Spectral Density (PSD) function is a particularly relevant transformation for resting-state EEG studies, where the powers of different frequency bands can be investigated. This can be used for short simulation times (several seconds), and it can be calculated for an individual NMM node, so training can be stored within GPU memory even with a large batch size. However, this involves having a FFT as part of the objective function which is relatively complicated operation for a backpropagation, and it is unclear to what level of success it could achieve. Thus, we present results of using a PSD as an objective function for a simple sine function, and then on an individual RWWEI node.

We observed that the PSD objective function (see Table 4.2) succeeds at optimizing the amplitude parameter of a simple sine function, but has difficulty fitting the frequency term. In Figure 5.3, the successful training of amplitude using the PSD objective function for various initial parameter values is shown. These results are evidence that the PSD objective function has a good gradient to follow, at least when optimizing amplitude. However, fitting frequency or phase did not perform well. In fact, for successful amplitude fitting, the frequency must be very close to the target frequency, although the phase can be different. The fact that the phase can be different is a benefit because the purpose of the PSD objective function is to be able to fit to resting-state data, for which there is no phase reference (such as a stimulation time) between the model and empirical data to align with.

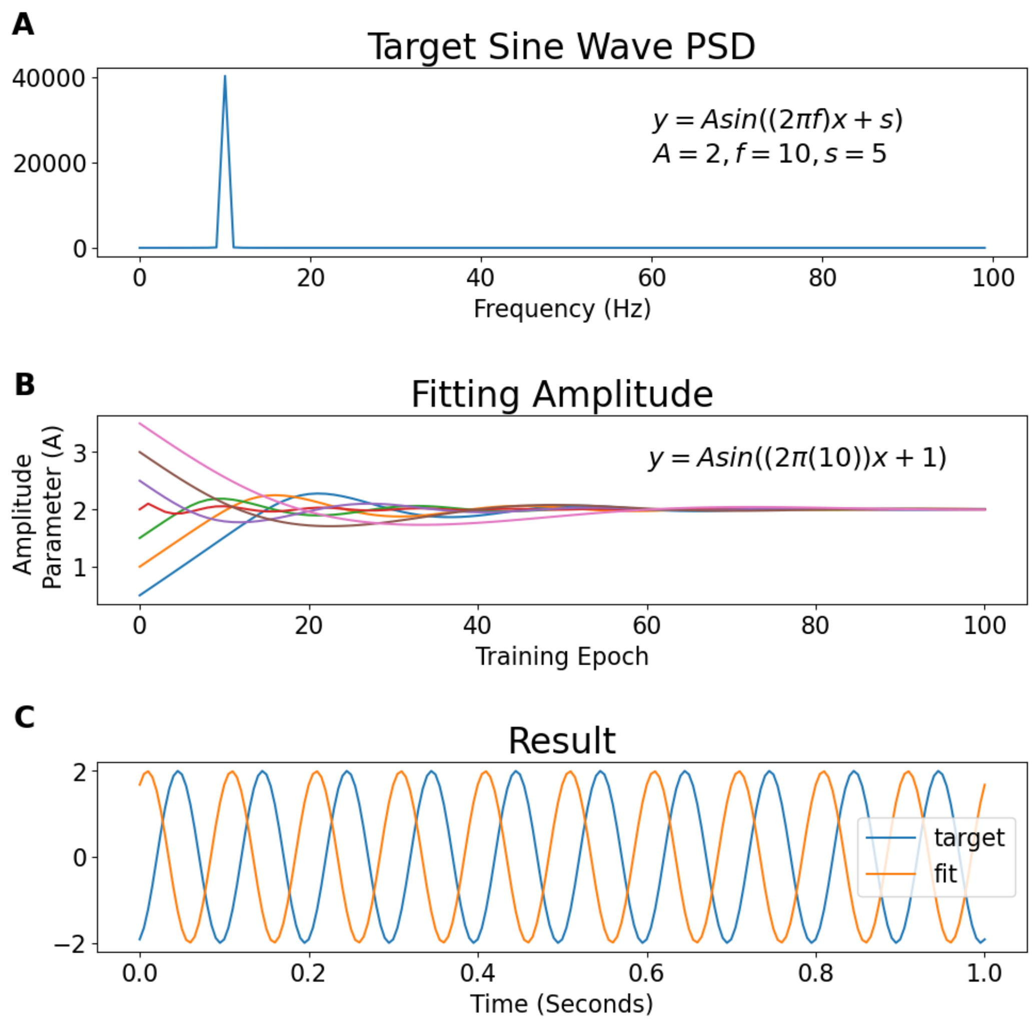

Figure 5.3 – *Fitting the sine wave amplitude (A) parameter using a PSD objective function.* **Panel A:** the target PSD. **Panel B:** each line represents an independent training run where the amplitude parameter being fit is started at different values. All training runs trended towards the optimal value of 2. **Panel C:** fitted sine function vs the sine function used to generate the target PSD.

Next, the PSD was evaluated when applied to an individual RWWEI node. To replicate the successful scenario with the sine function, a parameter which modulates the oscillation amplitude with a relatively consistent frequency is required. It was observed in Chapter 3 that beta (12-30 Hz) was a relatively robust oscillatory frequency in the model. Furthermore, we observed a local optimum for maximizing amplitude of oscillations (Figure 3.7 & Figure 3.8). A maximum amplitude of oscillations is observed near $I_{external}$ = 1. Thus, the $I_{external}$ variable is chosen to be fit as there is a known local optimum for trying to maximize the amplitude of beta frequency. In Figure 5.4, the parameter sets from Table 3.3 are used to see if BPTT using the PSD objective function finds the desired gradient in an RWWEI node.

Multiple variations were attempted to improve the PSD objective function gradient, by progressively adding more details (seen in Figure 5.1) to make the problem easier for the training algorithm to succeed. This resulted in specifically designed training parameters, and an objective function processing that matched closely with how the target PSDs were generated. This was to guarantee that the model was capable of fitting the target PSD very closely with close to zero loss.

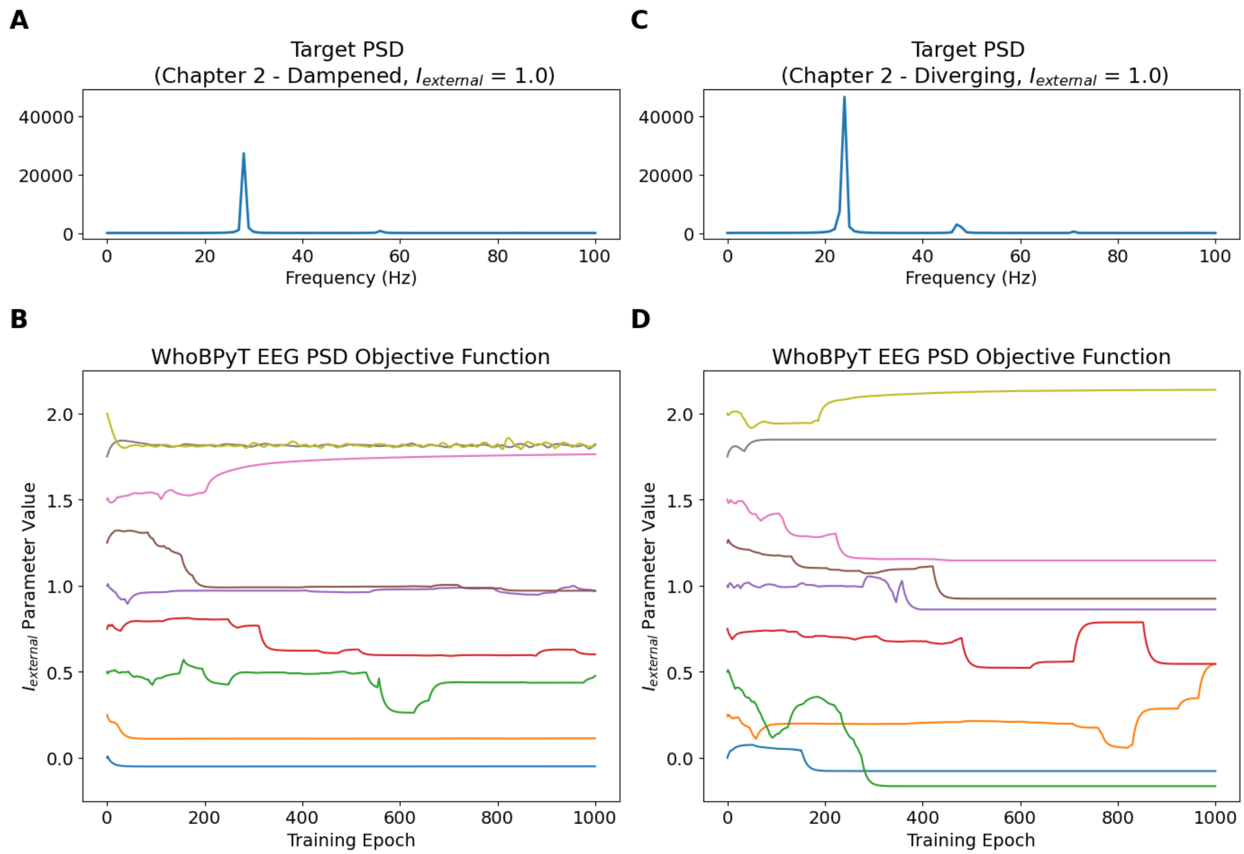

Figure 5.4 – *Fitting an individual RWWEI node using the PSD objective function.* Panels A&C: Shows the target PSD generated using the two parameters sets with alpha oscillations from Chapter 3 (dampened case has additive noise, diverging case does not have noise). **Panels B&D:** Each line represents a different training run with a different initial value for the $I_{external}$ parameter. The optimal value is around 1, which can be seen by the analysis from Chapter 3, for which the other model parameters came from. The training algorithm was not successful in these experiments, as the parameter values did not trend toward the known local optimum value.

Training Specifications	
CNMM Model	RWWEI2
Objective Function	MSE PSD
Optimization Paradigm	Fitting Batch
Parameters fit	$I_{external}$
Number of epochs	1000
State Value Initialization	Continued from previous backpropagation
Simulation time	Batch of 120 with 1 second each, for each of 10 disconnected nodes
Optimizer	Adam
Learning Rate	0.01
Device	GPU
Run Time	~15 hours/fitting

Table 5.1 – Training specifications for PSD objective function.

The target data was a synthetic PSD generated from the RWWEI verification implementation at a known local optimum parameter set, with respect to $I_{external}$. This was to ensure that a good fit could be achieved. For this synthetic PSD, 120 isolated nodes (connectivity matrix of zeros), run for 1 second to remove the initial condition transient, then run again for the recorded 1 second. The mean was subtracted from each 1 second segment, and then the PSD calculated and averaged. This is similar to Bartlett's method with the exception that each segment is independent. By design, this has one PSD point for each frequency along the x-axis, which makes the PSD less noisy but also high enough resolution to focus on particular frequencies.

The training in the RWWEI PyTorch implementation used a simulation length of 1.5 seconds with a batch size of 120. The objective function was set to discard the first 0.5 seconds as transient, then calculate the average PSD across the batch of 120. To further exploit GPU speed, this was performed for 10 nodes which had all zeros as connection strengths. So, the target PSD was repeated for each node, and the error calculated across all nodes. Finally, the range of 4-70 Hz was focused on, before calculating the MSE between the simulated PSD and target PSD. The model was trained for 1000 epochs (backpropagations), where the state variables continue instead of being reset. Essentially, this is trying to fit the $I_{external}$ for an individual node. Results of these analyses can be seen in Figure 5.4.

Despite this careful processing to make the simulated and target PSD processing steps very similar, it proved to be a difficult task to optimize the oscillations of a RWWEI node using the PSD objective function. This is seen by contrasting Figure 5.3 and Figure 5.4, where the fitted parameter values do not converge to the known local optimum value for independent fittings in the second case. This suggests that the objective function does not have a smooth gradient to follow. That is, there could be many local optima. Perhaps this is due to the slight change in frequency that is also occurring. For now, the brute force technique of Chapter 3 is the currently recommended technique used to find a parameter set for which the model oscillates.

5.3.2 Fitting the Global Connectivity and Noise Magnitude Parameters

One of the most fundamental parameters of a CNMM, particularly to tune for optimizing FC, is the global coupling strength, denoted G. This parameter scales the input from the other nodes via the SC matrix. It has been shown to have a local optimum value for the SC to simulated FC correlation, as well as for simulated FC to empirical FC correlation (Deco et al., 2014). The optimal value of G may depend on the chosen magnitude of noise (σ) for the model.

To identify a local optimum for the parameter G and determine the ground truth task gradient, a grid search was performed trying a range of G and σ values. A similar analysis had been performed for the RWWEI model in the past (Deco et al., 2014), which showed an initial increase in correlation with an increase in G. To simplify the analysis here, synthetic SC was used for both the SC and target FC. This was expanded to also show a dimension for changes in the noise magnitude, represented as a heatmap. For each combination of G and σ the FIC method was performed to tune the local J_i parameter values. The correlations between simulated FC and target FC were calculated for 3 runs and the median correlations are presented as a heatmap in Figure 5.5 Panels A&D for BOLD and source space EEG.

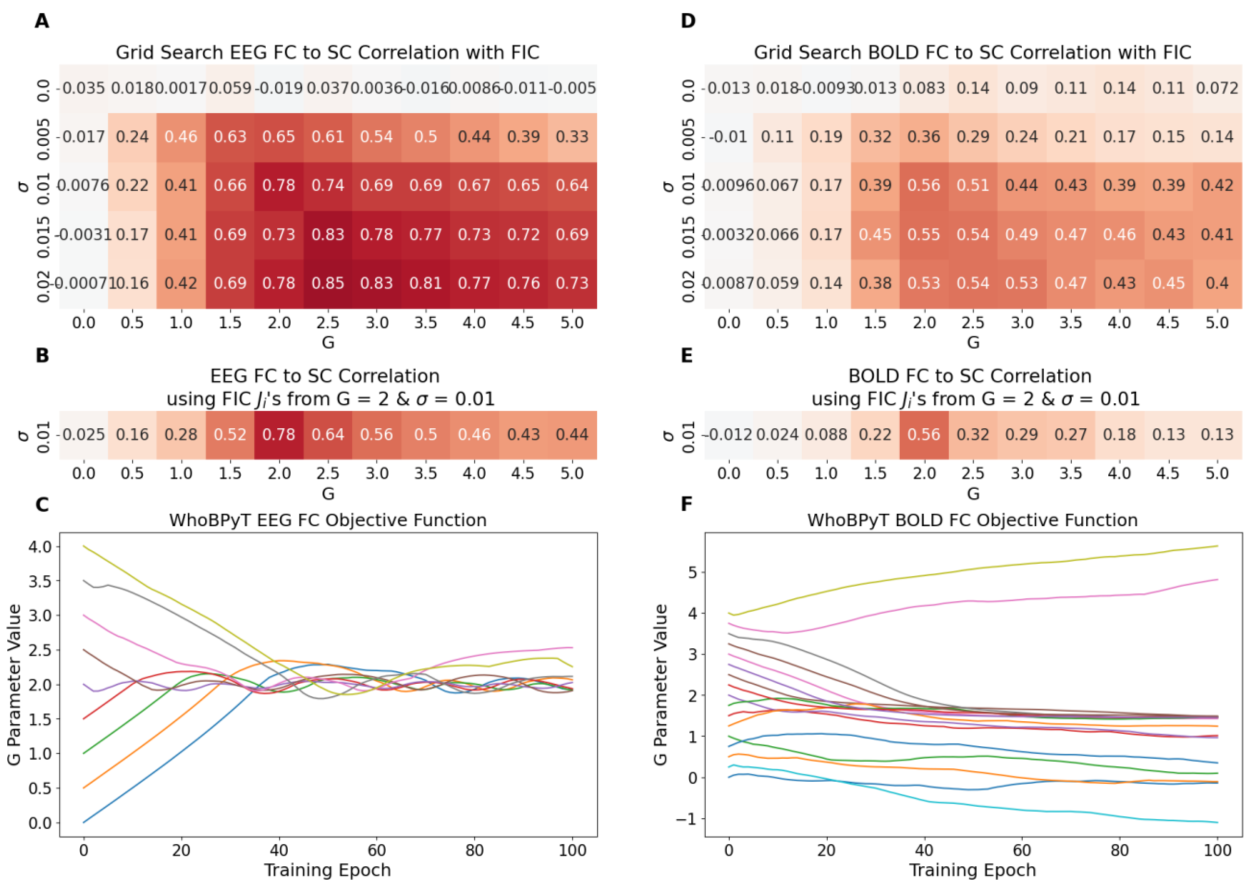

Figure 5.5 – *Fitting Source EEG FC and fMRI BOLD FC to a synthetic SC.* Panels A&D: Heatmap of Source EEG FC to SC and source BOLD FC to SC. Each value calculated is part of a grid search for G and σ, where for each combination of values WhoBPyT was used with a Mean S^E objective function to perform FIC. **Panels B&E:** Similar heatmap to A & D but using the optimal FIC fit parameters from σ = 0.01 and G = 2 and performing a sweep of G. **Panels C&F:** Optimizing the G parameter with WhoBPyT using different starting values with an EEG based and a BOLD based objective function. The EEG FC objective function is following the ground truth task gradient presenting in Panel B, but BOLD FC objective function is not showing clear success at following the Panel E gradient.

Training Specifications	
CNMM Model	RWWEI2
Objective Function	Average Local mean S^E Value
Optimization Paradigm	Fitting Batch
Parameters fit	J_i for all 80 nodes
Number of epochs	100
State Value Initialization	Reset after each backpropagation
Simulation time	0.1 seconds by 300 batch
Optimizer	Adam
Learning Rate	0.05
Device	GPU
Run Time	~7 minutes/fitting

Table 5.2 – Training specifications for performing FIC for grid search heatmap.

Training Specifications	
CNMM Model	RWWEI2_EEG_BOLD
Objective Function	BOLD FC or EEG FC
Optimization Paradigm	Fitting FNGFPG
Parameters fit	G
Number of epochs	100
State Value Initialization	Continued from previous backpropagation
Simulation time	30 seconds in 1 second blocks
Optimizer	Adam
Learning Rate	0.05
Device	GPU
Run Time	~24 hours/fitting

Table 5.3 – Training specifications for fitting G with known local optimum.

This grid search shows not only a smooth trend, but also a local optimum when dealing with just the G parameter. This is consistent with what would be expected from Deco et al. (2014). Furthermore, when selecting the best G value of 2 for the original σ of 0.01, then changing G while keeping the FIC J_i values fixed still maintains the gradient with a local optimum (see Figure 5.5 Panels B & E). This is important, as it suggests that backpropagation could have a good gradient to follow during training. Furthermore, this property provides a ground truth scenario that is very useful for evaluating the performance of the algorithm. For comparison, some analysis was also run without FIC, which found local optimums with lower correlations of around 0.58 and 0.33 for EEG and fMRI respectively (compared to 0.85 and 0.56 in figure 5.5) and occurring closer to G = 0.5.

In Panels C & F of Figure 5.5, only G is fit using the parameters corresponding to G = 2 and σ = 0.01, where the G parameter is trained from different initial values to see if it returns to the local optimum, which might also be the global optimum. In the case of EEG FC, the algorithm correctly finds the desired G value from various initial parameter values. However, for the case of BOLD FC, the G parameters may deviate away from the local optimum, indicating that the method is not working as intended.

One possible reason why the EEG FC objective demonstrates clear success while the BOLD FC does not may be the longer time scale, as RNNs are known to have issues with exploding or vanishing gradients for long input sequences. It is possible that the FNGFPG approach helped with this, but that contribution is difficult to isolate, as it is impractical to run a comparison analysis using the true-time scale BOLD with a standard fitting paradigm.

5.3.3 Maintaining EEG Oscillations while Improving BOLD FC

Now that we have examined the results of fitting to EEG and fMRI modalities individually, we will move on to multimodal parameter fitting tasks next. As was observed in Section 5.3.1 and Section 5.3.2, backpropagating through the MSE between the simulated and empirical PSD, and additionally the true-time scale BOLD FC parameter fitting, has so far not shown clear success in the RWWEI experiments of this study. Thus, a simplified problem for evaluating the multimodal case was performed. We have a successful technique for finding parameter sets for which the model oscillates in Chapter 3, and furthermore we observed that performing FIC improved BOLD FC to SC correlation in Section 5.3.2. Thus, a natural experiment to do was to start with a parameter set for which the model oscillates and then perform FIC starting within this parameter regime. In this way a dominant frequency which shows a peak in the PSD can be chosen. Subsequently, it might be possible to improve FC by performing FIC while maintaining the oscillatory properties.

We used the Deligianni and Clayden (2016) supplementary information which is conveniently already in a format for others to easily reproduce the results from some analysis in this thesis, without requiring any neuroimaging processing. In this thesis, the WODI SC, beta FC, and BOLD FC were chosen as they had the highest mean correlations with each other across subjects (WODI SC to Beta FC Correlation: 0.278; WODI SC to BOLD FC Correlation: 0.191; beta FC to BOLD FC Correlation: 0.242). In this section, we take the average of SC and BOLD FC matrices across all subjects. During training, the SC matrix was normalized by dividing by the matrix 2-norm. An identity matrix was used as the LF, so the model was using source space EEG.

During training the initial conditions use WhoBPyT defaults which includes some randomness, in which case a node may not be oscillating and near zero. For evaluation, we used higher initial conditions to increase the chances of all nodes oscillating. As was observed in analysis from Chapter 3, where $I_{external}$ is interpreted to be the net input from the rest of the network, it is possible for a NMM node to be stuck at the lower non-oscillating fixed point in a hysteresis. This can explain why the oscillations appear to be slightly above the target mean firing rate. One reason why performing FIC (here using Mean S^E) might be expected to improve performance is that it may be moving the fixed points of the system all around to the same value. This makes choosing appropriate initial conditions near fixed values a simple task, compared to determining a set of fixed points for the network before performing FIC.

With default parameter the network was not oscillating and simulated to empirical BOLD FC correlation was 0.067, with parameters for which the model oscillates it became 0.321, then after performing FIC with WhoBPyT it became 0.031. During this evaluation we used higher initial conditions of 0.2 for all nodes, instead of the WhoBPyT default, so that every node in the network was oscillating. An example simulation with the final parameters and the associated PSD and FC is shown in Figure 5.6.

Interestingly, using parameters for which the model oscillates appears to improve FC performance, but contrary to expectations enforcing biologically constrained mean firing rates reduced the simulated

FC to SC correlation in this parameter space. Whereas in the previous section of correlation FC to SC from the default parameter set, the FIC improved correlation, for this oscillating parameter set on empirical data the FIC objective function instead decreased correlation.

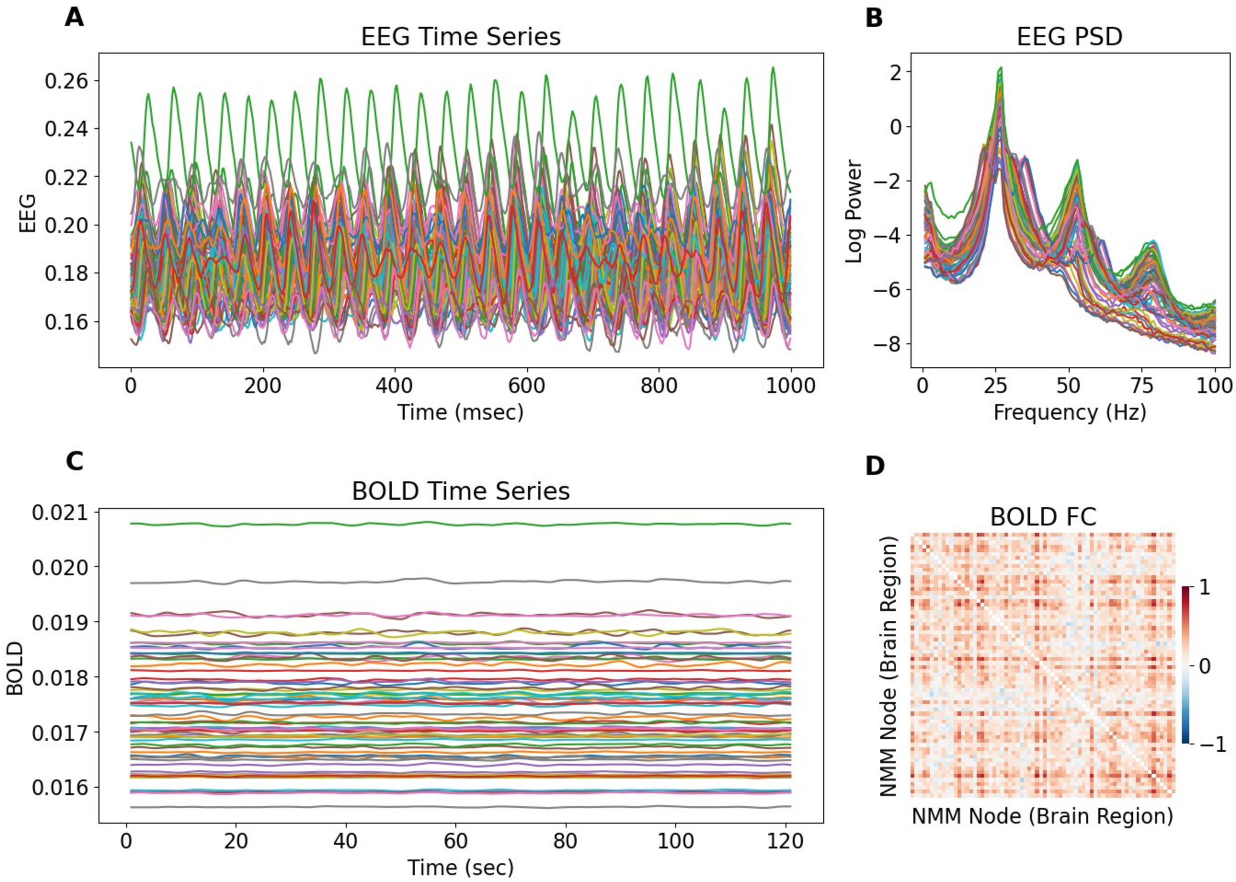

Figure 5.6 – *Full Multimodal RWWEI Numerical Simulation.* Panel A: Segment of EEG simulation**. Panel B:** Simulated EEG PSD. **Panel C:** Segment of BOLD simulation. **Panel D:** Simulated BOLD FC. Parameters selected from Chapter 3 for which the model oscillates and then FIC is performed using WhoBPyT. Even after training, the model still is in an oscillatory regime.

Training Specifications	
CNMM Model	RWWEI2_EEG_BOLD
Objective Function	Mean S^E
Optimization Paradigm	Fitting FNGFPG
Parameters fit	Local J_i values (J_{NMDA}, J, J_{new}, w_+, W_E, W_I already fit globally)
Number of epochs	500
State Value Initialization	Reset after each backpropagation
Simulation time	2.0 second with 0.1 second blocks
Optimizer	Adam
Learning Rate	0.02
Device	GPU
Run Time	~7 hours

Table 5.4 – Training specifications for performing FIC on model starting in oscillating parameter regime.

5.3.4 Evaluating the Benefit of Multimodal Fitting

A fundamental question of multimodal parameter fitting is whether better model performance can be achieved after fitting to multiple modalities simultaneously than fitting to one modality individually. Perhaps, even if only one modality is being studied, extra information from another modality could help find a better parameter regime to study biological questions. Our hypothesis, assuming the model, data, and learning algorithm are all sufficiently good, was that fitting to EEG and BOLD simultaneously would lead to better performance than fitting to either EEG or BOLD individually. We tested this hypothesis with the RWWEI model, on the (Deligianni et al., 2016) supplementary materials dataset, using WhoBPyT with various objective functions. The test determines whether when fitting to both source space EEG FC and fMRI FC, the model outperforms fitting on either source space EEG FC or fMRI FC individually.

5.3.4.1 Fitting EEG FC and fMRI BOLD FC

The parameters chosen to be fit were the global G and σ parameters and the local J_i parameters. These are parameters conventionally used in the literature when trying to fit BOLD FC, though the possibility of fitting additional parameters has already become easier with current WhoBPyT functionality. The global G and σ were bounded below by zero using the log feature. Fitting FIC may lead to better FC correlations, as was the case for synthetic data in section 5.3.2, and also put the model's firing rates in a more biologically relevant range, and helps prevent any J_i parameter from going negative, so the mean S^E objective function component (for performing FIC) is added to the objective function in all cases in this section.

The hyperparameters were chosen such that they would be expected to work across all objective functions. Longer simulations were backpropagated through to accommodate fitting BOLD FC, even though for the case of only fitting mean S^E it would be more efficient to use shorter time lengths. Thus, simulation lengths of 10 seconds in FNGFPG with 1 second blocks were used. This block size is similar to an fMRI TR, and training could be performed at a rate of roughly 5 minutes per epoch. Additionally, state variable initial conditions were not reset after each backpropagation to accommodate the longer transient period of BOLD (which is more computationally efficient than removing a transient period from the slow time scale each backpropagation). Whereas, for the case of only fitting mean S^E it would have made sense to reset the initial conditions to the target value after each backpropagation (where the target steady state is known, and there is a fast time scale transient period). For each training, fitted model parameters always started at the same default values. Other details on the exact algorithm structure can be found by looking at the v0.2.0 code of the WhoBPyT repository.

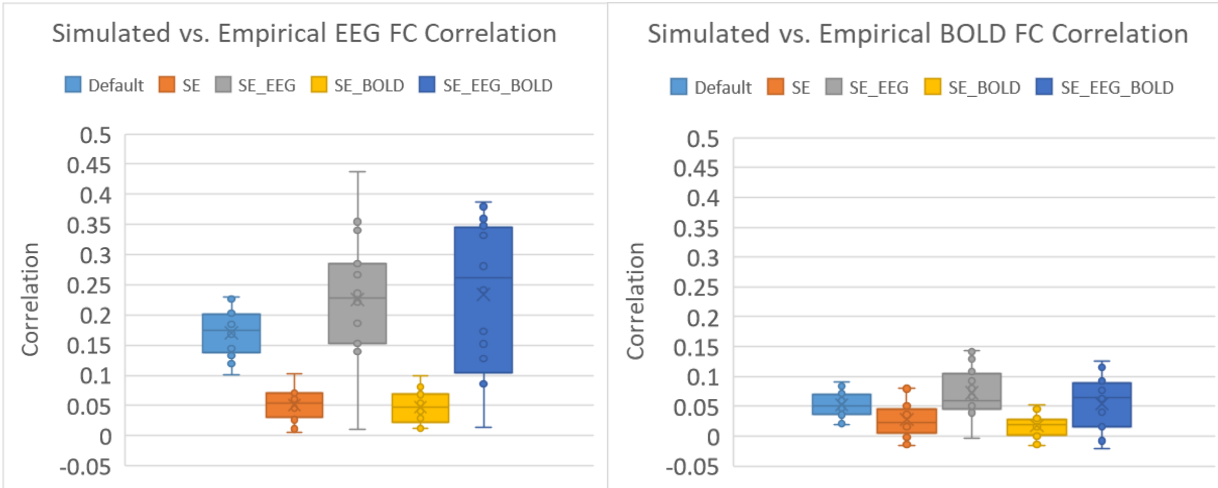

Figure 5.7 – ***The performance of source space EEG FC and fMRI BOLD FC for the default parameters and four objective functions.*** Training was performed on 16 subjects for each objective function. For this dataset, Mean S^E for FIC reduced performance, but the EEG FC objective function component seemed to show a trend of improvement. Left plot is for source space EEG and the right plot is for BOLD. (Created with Excel)

Training Specifications	
CNMM Model	RWWEI2_EEG_BOLD
Objective Function	Various
Optimization Paradigm	FNGFPG
Parameters fit	G, σ, J_i
Number of epochs	200
State Value Initialization	Continued from previous backpropagation
Simulation time	10 seconds with 1 second blocks
Optimizer	Adam
Learning Rate	0.05
Device	GPU
Running Time	~17 hours/fitting

Table 5.5 – Training specifications for comparing individual modality vs. multimodal fitting.

Evaluation simulation time was 5 minutes, with step sizes of 0.1 milliseconds. Initial conditions were generated using the method provided by the model class. The initial transient period of 30 seconds was removed. Before calculating FC, EEG was down-sampled selecting every 20th step, fMRI was down-sampled by selecting every 1000th step. The empirical EEG and fMRI precision matrices were inverted to get the empirical FC matrices, for which the lower triangle was correlated with the corresponding simulated FC matrices. In Figure 5.7, the correlations between simulated and empirical FC are plotted for the various objective functions.

When working with empirical data where the optimal parameters are unknown, a different baseline needs to be used to evaluate performance. To evaluate baseline Source EEG FC performance, the correlation between SC and Source EEG can be used. For the BOLD FC baseline performance, the correlation between SC and fMRI BOLD FC can be used. In this way, we can determine whether the CNMM model dynamics are operating in a way to generate meaningful output. In Figure 5.7, some subject runs showed above baseline EEG FC performance, but none showed above baseline BOLD performance.

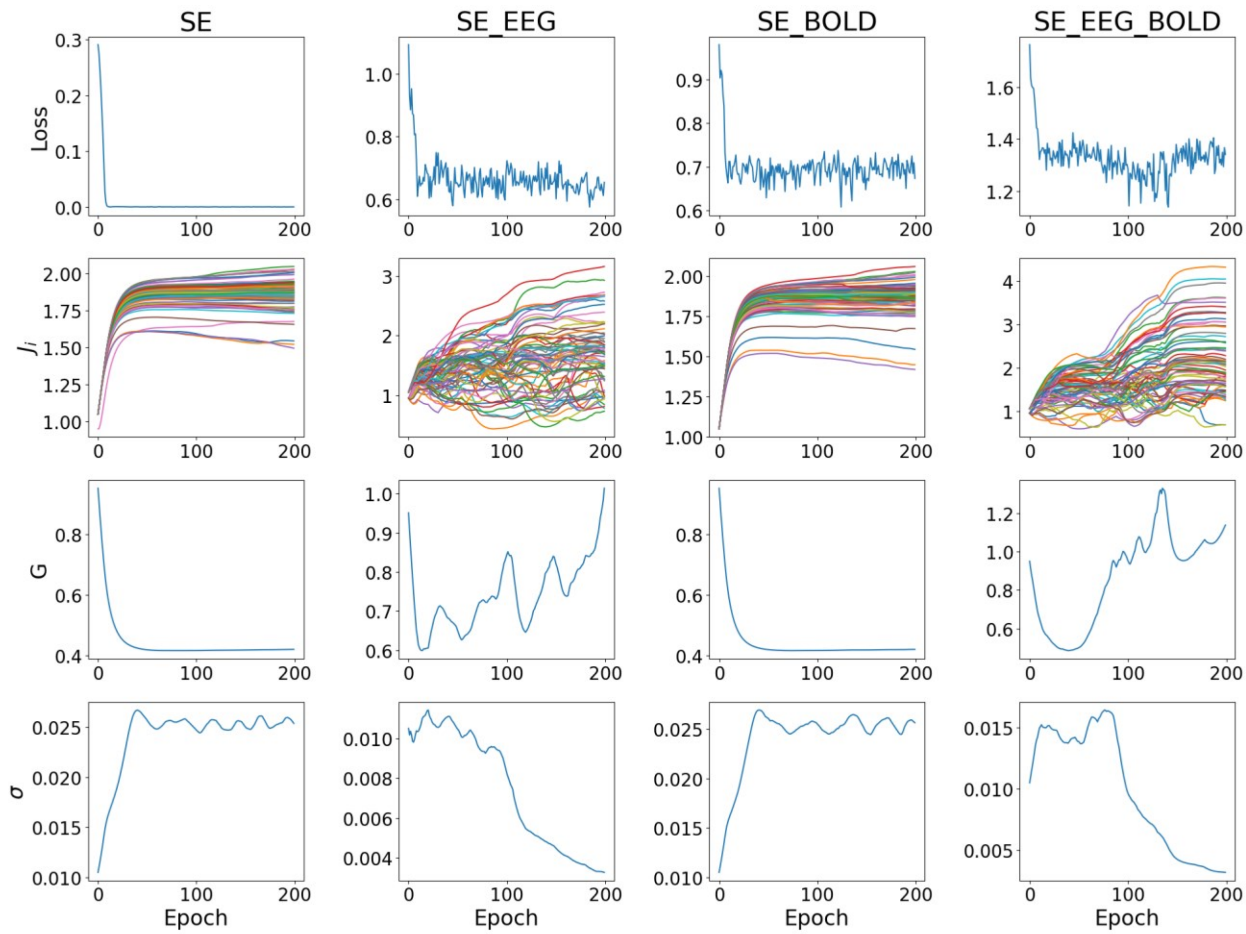

Figure 5.8 – *Example training metrics for one subject across four different objective functions. Training loss (first row) and parameter trajectories (second to forth row) are presented here.* For these training of one subject, an initial rapid decrease in loss due to performing FIC well is observed, but it is less clear whether a slower trend with improved FC is present. Also, for these trainings, the parameters stabilized for the SE and SE_BOLD objective functions, but not for the objective functions with an EEG FC component.

We examined the parameter values after training for each of the objective functions. An example from one subject is shown in Figure 5.8. What we would like to see is an initial decrease in loss due to the easier Mean S^E task, and then a slower decrease due to the objective function components based on FC. Furthermore, it would be preferred that the parameter trajectories stabilize to a certain value as the loss decreases.

Upon examining the parameter trajectories across subjects, the SE and SE_BOLD objective functions had similar and consistent behaviour. This suggests that the optimizer is not finding a consistent gradient to follow for the BOLD FC objective function component. On the other hand, the SE_EEG and SE_EEG_BOLD objective function commonly showed an initial decrease in G and increase in J_i similar to the other objective functions, after which there is a variation was observed across subjects in parameter trajectories. The noise term for some subjects decreased to near zero which is a problem for FC performance, because without noise the model may simply dampen to steady state. Thus, treating σ as a hyperparameter to be tuned rather than a parameter to be optimized by backpropagation may make training more robust.

5.3.4.2 Comparison to Another Implementation

To inquire into potential reasons that the BOLD FC performance obtained had poor correlation values, we investigated the performance in a different implementation using Neurolib. Neurolib is convenient to use as it has a simple interface, with an implementation that is easier to examine and modify, if necessary, compared to larger packages such as TVB. It uses Numba and a chunking technique that in combination allows for the generation of longer BOLD time series and is optimized to run fast on CPU. The subsequent Neurolib results were performed using simulations and metric calculations were run on GitHub Codespaces with 4 Cores and 16GB RAM.

To make the Neurolib simulations as close as possible to the deco et al. (2014) equations and parameters like in the WhoBPyT RWWEI2 NumPy implementation, some specific parameter values were chosen. With the Neurolib code from git commit #4436468 (which has a bug fix that was identified in the process of performing this analysis), the following parameter choices were made:

- `tau_ou` was set to match the step size of 0.1, which for this implementation has the effect of removing the Ornstein-Uhlenbeck process and instead having additive white noise.
- `sigma_ou` was set to 0.1 to be equivalent to the original noise of 0.01 divided by the step size, due to the noise being added to the derivative term before being multiplied by the step size.
- The `default_output` was set to "se" which would be the input to the BOLD equations.
- The global coupling `K_gl` was set to 1.0.
- The SC matrix was normalized by the matrix 2-norm before being provided to Neurolib.
- The distance matrix was set to all 0's to remove delays.
- BOLD time points were saved at a rate (TR) of 1 per second.

As seen in the upper plots in Figure 5.9, there is a clear relationship between Neurolib and WhoBPyT for EEG FC correlation performance, but the expected relationship for BOLD FC correlation performance was not present (though the means are similar). This led to further investigation using Neurolib.

In the main section of Figure 5.9 we examine the model behaviour with the default parameters compared with a slightly improved G value (predicted based on Section 5.3.2 observations). These experiments were designed so that the analysis could be performed without modifying Neurolib (which currently does not have support for local J_i values). The finding is that for default parameters, the correlation between BOLD FC matrices from independent simulations is quite poor, although it does improve with increased simulation length. This explains why the WhoBPyT BOLD FC performance would not have a strong relationship with a rerun in Neurolib. As it turns out, though we see a change in the mean performance between objective functions, individual subjects show a large variation when comparing two independent simulations with the trained parameters when both simulations are performed with WhoBPyT.

We observe that choosing a global G value of 0.5 in the non-FIC case results in improved correlation of the simulated FC to empirical FC across subjects. This furthermore leads to even greater improvements in terms of the model's BOLD FC correlation to SC (which relates this analysis to Figure 5.5) and furthermore with itself. This may be a contributing factor to why fitting BOLD FC seems to be a more challenging task than fitting EEG FC. If the parameters are not within some optimal range, then the model may not produce a reliable gradient via backpropagation for which to follow during gradient descent.

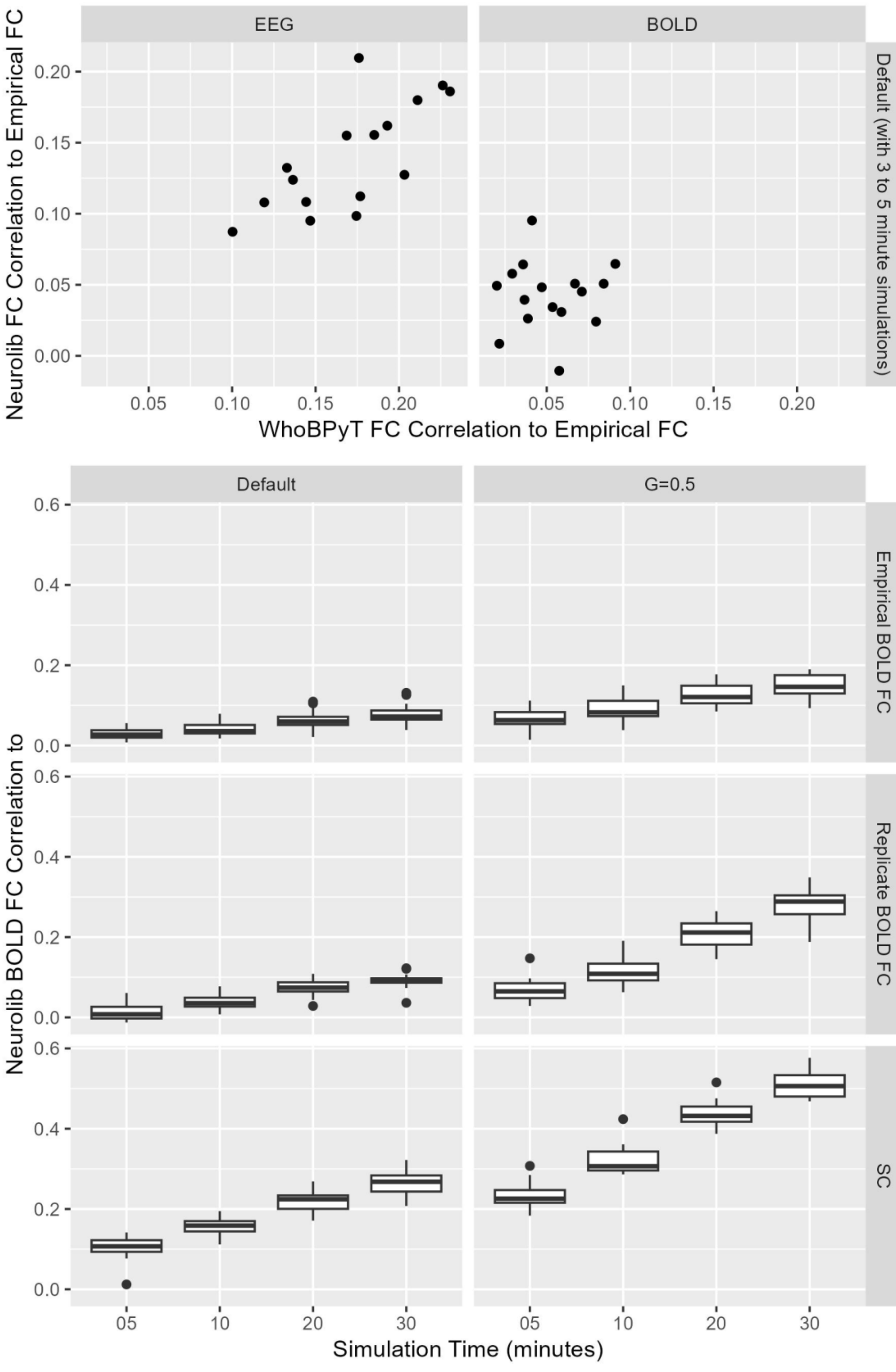

Figure 5.9 – ***Comparing results from simulations in WhoBPyT Validation models vs. Neurolib simulations, as well as longer Neurolib BOLD Simulations to itself.*** We observe that for a better G parameter value to reproduce the Empirical BOLD FC, the model simulated FC is more correlated with a rerun. (Created with Tidyverse)

5.4 Discussion

5.4.1 Evaluation Summary

This chapter performed multiple experiments working towards the goal of fitting CNMM parameters to reproduce concurrent EEG and fMRI neuroimaging phenomena. In particular, we developed and explored a PSD objective function for resting-state EEG, an FNGFPG optimization paradigm for fitting true-time scale BOLD, and a multimodal objective function for fitting EEG and fMRI BOLD concurrently. There were various degrees of success.

The PSD objective function showed promise on the sine function but not yet on the NMM. The focus was on adding processing details to make the simulated and target PSD as close as possible. Additionally, fitting paradigms were exploited to try to get a smooth PSD and use a batching technique. Regardless, it still did not show clear success, and this difficulty could be a result of many local optima, particularly when trying to fit frequency.

The FNGFPG paradigm allows for much longer simulations to be backpropagated through. However, it does not achieve as big of a speed boost, compared to exploiting the batching paradigm for the PSD objective function, because it still needs to run the serial simulation to get a set of initial conditions. The paradigm was shown to be compatible with various objective functions, but unfortunately, it was not clear whether it helped with BOLD FC which was the main motivation of its creation. This paradigm is also limited in that it would not work well if the user wanted to integrate delays into the model.

In previous work by the CAMH WBMG, the success included fitting SC and LF weights on modified models, which was not done here to put more emphasis on the NMM itself. This made it a more challenging problem. There are many more constraints in CNMMs than in CTRNNs, so they might not be a universal function approximator. Currently, the parameter fits are specific to an individual subject which have their own SC matrix (and possibly also the LF matrix) based on experimental measurements.

These evaluations are not just useful for evaluating WhoBPyT's performance, but also for calibrating hyperparameters on simplified problems as a starting point for working on more advanced ones. That is, what learning rate and how many epochs are required to achieve certain performance, as well as the run time to complete analysis. Interestingly, the best BOLD FC performance so far was achieved by using parameter sets for which the model oscillates without further tuning in Section 5.3.3, so perhaps it would be worth looking into this further.

The calculations are very computationally intensive, and they are similar whether fitting one parameter or many parameters, so it is more practical to use this methodology in the case of fitting many parameters. In practice, when fitting only a few parameters, a grid search may be more efficient and have superior performance. However, the benefit of the gradient descent via backpropagation-based technique is that it can explore a much larger parameter space without the exponentially increasing in run time that a grid search would require. The reason only a few parameters were examined is for the purpose of evaluating the algorithm by comparing the found parameters to known ground truth local optimums.

5.4.2 Limitations of Results

The results here have various limitations, in terms of generalizability. The ideal scenario is to have a methodology that can fit an arbitrary set of parameters in a generic CNMM to reproduce various phenomena and characteristics from any chosen dataset. Of course, this is unlikely to happen in practice, so experimentation and tuning will still be required.

The analyses of this chapter were simplified in multiple ways, due to various factors and constraints. In particular, the phase relationship between EEG and fMRI was dropped by focusing on fixed target FC and fixed target PSD objective functions. This does not fully utilize the benefit of having concurrent resting-state EEG and fMRI data, where the phase relationship between modalities is what makes it unique compared to doing separate recordings. Additionally, EEG was simplified from what was done in previous work by using source space EEG, which did not require creating a LF matrix. Finally, a simplification of staying with optimization paradigms instead of expanding into machine learning ones, where parameters would generalize across subjects. These simplifications could be added back in future work.

Though some results seem quite robust – as suggested by similarities across findings in the literature, analysis on synthetic data, and analysis on empirical data – other results might be biased and more dependent on the data used. In particular, the SC could play a big role in the trajectories of parameters, and furthermore, the quality of the empirical recordings and processing could have a big impact on what parameter values are optimal and the performance that that model could achieve.

Similarly in terms of the models, perhaps a different NMM would result in different conclusions. In particular, the PSD objective function, which appears to work on a sine wave function but not on the RWWEI model, would succeed for a different NMM. Or perhaps, it could work for the RWWEI model as well, but for a different local optimum fitting a parameter other than $I_{external}$.

The Adam optimizer provided in PyTorch was used for all experiments. This optimizer is more advanced than the SGD option, so it is often preferred. However, for the purpose of evaluating the gradient of an objective function, it can follow a trajectory that is not consistent with the local gradient. Granted, there is noise in the system, so an individually computed gradient may not be smooth for the SGD optimizer either. Thus, for future work, it may be preferred to instead use the SGD optimizer and take the average gradient direction for multiple backpropagations, instead of analyzing the trajectory of the parameter across training epochs.

We have utilized machine learning techniques, but we have not followed machine learning paradigms. That is, though backpropagation is being used on a custom RNN, a separate model is being fit to each subject's resting-state data individually with noise as input (each subject having their own parameter sets). Thus, it would be more accurate to describe it as following an optimization paradigm. This means that statements cannot be made about the generalizability of found parameters to unseen data in the usual ML way. An idea of how standard ML paradigms could have been applied would be to, instead of considering noise as input into the model for the resting-state case, have the SC of the subject as the input and use the same target as before (i.e., based on PSD or FC of empirical time series). However, this has a challenge with the RWWEI model where the J_i parameters would be specific to the SC. That said, perhaps this balanced property of the model can be relaxed, as the best performance in this thesis was for a model without it.

5.4.3 Ideas for further improvement

The deep learning technique for CTRNNs used by WhoBPyT still has much more to be understood and evaluated. Much of the analysis applied to the RWWEI model in this thesis could also be applied to other NMM models, which have varying neural population representations. Furthermore, it would be beneficial to compare more combinations of how the different objective functions perform on different NMM and CNMM models (for various parameter combinations as well). This methodology has the potential to fit many more parameters, with some modification to the implementation of the equations. This has yet to be fully explored, such as evaluating combinations of local J_{NMDA}, J_i, J_{new}, w_+, W_E, W_I and other parameters. Some constraints such as to not fit the time constants were decided upon so as to adhere strongly to the biological assumptions, but it may be interesting to see if the algorithm can improve performance by fitting those parameters.

For the goal of fitting to multimodal data, which helps reduce the overfitting problem, more work needs to be done to improve the individual modalities. On the EEG side, a more biological EEG signal could be explored based on synchrony, possibly by using a different model or adding additional dynamics. The LF matrix could be added back to look at channel space EEG. Additionally, the PSD objective function could be further evaluated on other models, or perhaps instead an objective function component based on linear stability analysis could be used to promote a certain oscillation frequency independent of the empirical data. On the BOLD side, more work needs to be done to fit FC. Finally, adding connection time delays to the RWWEI model may make both modalities more biologically relevant, though it would likely require a different solution for the true-time scale BOLD problem.

There are some features that have not yet been explored that may help with training. So far, the weighting of various objective function components in this thesis have all used a value of one, where instead these can be considered hyperparameters which could be tuned. Another thing that could be done to improve performance is to select the parameters with the smallest loss value over training, instead of the parameters from the final epoch of training. This makes more sense when a validation dataset is used, and this was not done here as the focus was on evaluating the learning algorithm.

Reproducing the alpha-BOLD anticorrelation in an empirical dataset would be useful to know better the exact characteristics one should try to optimize. Part of the challenge of this is finding a public dataset in the required format and/or the complexity of the processing raw data required to prepare the data in the format used by CNMM models.

CHAPTER 6 – CONCLUSION

6.1 Summary

In summary, this thesis progressed the CTRNN BPTT methodology for optimizing CNMM parameters, extending it to the context of concurrent resting-state EEG and fMRI data. The fitted models are then useful for computational neuroimaging research applications, such as studying the alpha-BOLD anticorrelation phenomenon. With this in mind, it is of importance that the fitted models not only reproduce synthetic neuroimaging data well, but also do so in a biologically interpretable way. In this thesis, the RWWEI model was chosen to evaluate the deep learning methodology as well as for studying the alpha-BOLD anticorrelation phenomenon.

The studied parameter fitting methodology is based on theory from the 1980's that has recently become much more convenient to implement due to increased computer power and availability of tools such as PyTorch. The gradients were calculated using BPTT, treating the CNMM itself as a CTRNN for parameter fitting. The methodology was evaluated for various combinations of fitted parameters and objective functions, on whether the found parameters matched the known local optimum from brute force methods, or whether the model's ability to reproduce the desired phenomena improves.

We examined the oscillatory and FC characteristics of the RWWEI model, as these are time invariant phenomena, which we can fit the model to in the case of resting-state EEG and fMRI. Linear stability analysis was used to explore the oscillatory behavior of an individual RWWEI node, which allowed for a more efficient brute force search of frequencies present over a range of local coupling values and base inputs. For FC, simulations were still required to be run in a grid search. These were brute force search techniques, for which a limited number of parameters can be extensively explored. When moving on to fit more parameters in CNMMs, alternative methods such as the backpropagation technique become necessary.

Several contributions and successes were made. The WhoBPyT Python package was progressed closer to a release state. Local RWWEI parameters were successfully fit to set a target mean firing rate, as a deep learning alternative to the original FIC technique. With regards to oscillatory properties, a linear stability analysis technique for finding innately oscillating RWWEI parameters worked efficiently. Though the new PSD based objective function was shown to have a good gradient for optimizing amplitude, experiments applying it to an individual RWWEI node did not work convincingly. With regards to FC correlations, success was shown for EEG but less so for fMRI BOLD. True-time scale simulations made backpropagating through longer time simulations more computationally efficient, and is expected to reduce the exploding/vanishing gradient problem of RNNs. Regardless, the BOLD FC correlation did not show convincing results. As such, more work needs to be done for the individual components of the multimodal objective function to work better before truly evaluating whether the multimodal objective function outperforms the rest.

The chosen application to study was the alpha-BOLD anticorrelation. This was because it is a phenomenon relating the EEG and fMRI modalities together. We worked towards being able to optimize parameters at the local node level so that they can be analyzed and compared using linear stability analysis techniques to come up with novel candidate mechanisms. This could then compare parameters from brain regions which have positive vs. negative alpha-BOLD correlations. For now, an individual RWWEI node was studied in parameter regimes for which the model oscillates, to come up with two candidate mechanisms for the alpha-BOLD anticorrelation phenomenon.

This code was contributed to as additions to the WhoBPyT Python package. Before being able to incorporate this new code, the WhoBPyT Python package needed to be further developed from a software engineering perspective. Much is still to be implemented, but regardless many more experiments can be performed with the currently available functionality. In fact, the package has been specifically designed to conveniently perform similar experiments on many different parameter combinations.

6.2 Future Work

The WhoBPyT code base is ready for more convenient experimentation going forward. The experiments so far have focused on parameters that have traditionally been fit. However, with WhoBPyT there are many different parameter combinations which can be conveniently experimented on and have yet to be explored. Furthermore, we could look for parameters that perform well across subjects, in a more standard ML approach.

WhoBPyT focuses on one particular deep learning technique, but there are many other approaches involving deep learning or machine learning techniques for parameter fitting. It may be useful for a comparison to be made regarding the performance of the WhoBPyT approach compared to other techniques. For example, the genetic algorithm provided in Neurolib.

It would be beneficial to compare the WhoBPyT CTRNN technique with other parameter fitting algorithms, where the biological and ML model are different models. These evaluations could use more standard ML paradigms where the data is split into training, validation, and testing, and the model is evaluated on its generalizing ability.

This methodology can be applied to other types of models in neuroscience as well, particularly network-based models, for example networks of individual neurons. Furthermore, there is more to be done in terms of the parameter optimization method, which can be applied to differential equation-based models outside of the context of WBM. It could be analyzed at the abstraction of different equation terms. There is still much to explore, in terms of which context or differential equation terms the backpropagation technique works well for.

Overall, what has worked so far when examining this technique for phase invariant phenomena is fitting the mean value of activity of the excitatory population as already shown in Chapter 4, optimizing the amplitude of a sine function using a PSD objective function, and fitting the G value for optimizing source space EEG FC. Currently, to get oscillations of a desired frequency a brute force approach using linear stability analysis from Chapter 3 is used, and to get BOLD FC either fitting FIC can be tried or fitting Source EEG and evaluating on BOLD FC is recommended.

More work should be done looking at concurrent resting-state EEG and fMRI at the time series level so that the relationship of the phase of the modalities can be examined.

BIBLIOGRAPHY

Almeida LB. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. IEEE First International Conference on Neural Networks, Sheraton Harbor Island East, San Diego, California. 1987 June 21-24 (pp. 609 - 618).

Aquino KM, Fulcher B, Oldham S, Parkes L, Gollo L, Deco G, Fornito A. On the intersection between data quality and dynamical modelling of large-scale fMRI signals. NeuroImage. 2022 Aug 1;256:119051.

Biswal B, Zerrin Yetkin F, Haughton VM, Hyde JS. Functional connectivity in the motor cortex of resting human brain using echo‐planar MRI. Magnetic resonance in medicine. 1995 Oct;34(4):537-41.

Brunel N, Wang XJ. Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. Journal of computational neuroscience. 2001 Jul;11:63-85.

Byrne Á, O'Dea RD, Forrester M, Ross J, Coombes S. Next-generation neural mass and field modeling. Journal of neurophysiology. 2020 Feb 1;123(2):726-42.

Cakan C, Jajcay N, Obermayer K. neurolib: A simulation framework for whole-brain neural mass modeling. Cognitive Computation. 2021 Oct 12:1-21.

Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. Advances in neural information processing systems. 2018;31.

Damoiseaux JS, Rombouts SA, Barkhof F, Scheltens P, Stam CJ, Smith SM, Beckmann CF. Consistent resting-state networks across healthy subjects. Proceedings of the national academy of sciences. 2006 Sep 12;103(37):13848-53.

David O, Kiebel SJ, Harrison LM, Mattout J, Kilner JM, Friston KJ. Dynamic causal modeling of evoked responses in EEG and MEG. NeuroImage. 2006 May 1;30(4):1255-72.

David O, Friston KJ. A neural mass model for MEG/EEG:: coupling and neuronal dynamics. NeuroImage. 2003 Nov 1;20(3):1743-55.

Deco G, Ponce-Alvarez A, Hagmann P, Romani GL, Mantini D, Corbetta M. How local excitation–inhibition ratio impacts the whole brain dynamics. Journal of Neuroscience. 2014 Jun 4;34(23):7886-98.

Deco G, Ponce-Alvarez A, Mantini D, Romani GL, Hagmann P, Corbetta M. Resting-state functional connectivity emerges from structurally and dynamically shaped slow linear fluctuations. Journal of Neuroscience. 2013 Jul 3;33(27):11239-52.

Deco G, Kringelbach ML, Arnatkeviciute A, Oldham S, Sabaroedin K, Rogasch NC, Aquino KM, Fornito A. Dynamical consequences of regional heterogeneity in the brain's transcriptional landscape. Science Advances. 2021 Jul 14;7(29):eabf4752.

Deligianni F, Centeno M, Carmichael DW, Clayden JD. Relating resting-state fMRI and EEG whole-brain connectomes across frequency bands. Frontiers in neuroscience. 2014 Aug 28;8:258.

Deligianni F, Carmichael DW, Zhang GH, Clark CA, Clayden JD. NODDI and tensor-based microstructural indices as predictors of functional connectivity. Plos one. 2016 Apr 14;11(4):e0153404.

Desikan RS, Ségonne F, Fischl B, Quinn BT, Dickerson BC, Blacker D, Buckner RL, Dale AM, Maguire RP, Hyman BT, Albert MS. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. Neuroimage. 2006 Jul 1;31(3):968-80.

Donoghue T, Haller M, Peterson EJ, Varma P, Sebastian P, Gao R, Noto T, Lara AH, Wallis JD, Knight RT, Shestyuk A. Parameterizing neural power spectra into periodic and aperiodic components. Nature neuroscience. 2020 Dec;23(12):1655-65.

Durstewitz D, Koppe G, Thurm MI. Reconstructing Computational Dynamics from Neural Measurements with Recurrent Neural Networks. bioRxiv. 2022:2022-10.

Effenberger F, Carvalho P, Dubinin I, Singer W. A biology-inspired recurrent oscillator network for computations in high-dimensional state space. bioRxiv. 2022:2022-11.

Elman JL. Finding structure in time. Cognitive science. 1990 Mar;14(2):179-211.

Feshchenko VA, Reinsel RA, Veselis RA. Multiplicity of the α Rhythm in Normal Humans. Journal of Clinical Neurophysiology. 2001 Jul 1;18(4):331-44.

Friston KJ, Harrison L, Penny W. Dynamic causal modelling. Neuroimage. 2003 Aug 1;19(4):1273-302.

Funahashi KI, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks. Neural networks. 1993 Jan 1;6(6):801-6.

Gaskin T, Pavliotis GA, Girolami M. Neural parameter calibration for large-scale multiagent models. Proceedings of the National Academy of Sciences. 2023 Feb 14;120(7):e2216415120.

Goldman RI, Stern JM, Engel Jr J, Cohen MS. Simultaneous EEG and fMRI of the alpha rhythm. Neuroreport. 2002 Dec 20;13(18):2487.

Griffiths JD, Bastiaens SP, Kaboodvand N. Whole-brain modelling: past, present, and future. In Computational Modelling of the Brain: Modelling Approaches to Cells, Circuits and Networks 2021 Oct 8 (pp. 313-355). Cham: Springer International Publishing.

Griffiths JD, Wang Z, Ather SH, Momi D, Rich S, Diaconescu A, McIntosh AR, Shen K. Deep Learning-Based Parameter Estimation for Neurophysiological Models of Neuroimaging Data. bioRxiv. 2022 May 19:2022-05.

Hasani R, Lechner M, Amini A, Rus D, Grosu R. Liquid time-constant networks. In Proceedings of the AAAI Conference on Artificial Intelligence 2021 May 18 (Vol. 35, No. 9, pp. 7657-7666).

Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

Honey CJ, Kötter R, Breakspear M, Sporns O. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. Proceedings of the National Academy of Sciences. 2007 Jun 12;104(24):10240-5.

Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural networks. 1989 Jan 1;2(5):359-66.

Huster RJ, Debener S, Eichele T, Herrmann CS. Methods for simultaneous EEG-fMRI: an introductory review. Journal of Neuroscience. 2012 May 2;32(18):6053-60.

Hutt A, Lefebvre J. Arousal fluctuations govern oscillatory transitions between dominant γ and α occipital activity during eyes open/closed conditions. Brain Topography. 2022 Jan;35(1):108-20.

Jackson AF, Bolger DJ. The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. Psychophysiology. 2014 Nov;51(11):1061-71.

Jaeger C, Nuttall R, Zimmermann J, Dowsett J, Preibisch C, Sorg C, Wohlschlaeger A. Targeted rhythmic visual stimulation at individual participants' intrinsic alpha frequency causes selective increase of occipitoparietal BOLD-fMRI and EEG functional connectivity. NeuroImage. 2023 Apr 15;270:119981.

Jansen BH, Rit VG. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. Biological cybernetics. 1995 Sep;73(4):357-66.

Jirsa VK, Proix T, Perdikis D, Woodman MM, Wang H, Gonzalez-Martinez J, Bernard C, Bénar C, Guye M, Chauvel P, Bartolomei F. The virtual epileptic patient: individualized whole-brain models of epilepsy spread. Neuroimage. 2017 Jan 15;145:377-88.

Kalat JW. Biological psychology. Cengage Learning; 2016.

Kidger P. On neural differential equations. arXiv preprint arXiv:2202.02435. 2022 Feb 4.

Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.

Klimesch W. EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. Brain research reviews. 1999 Apr 1;29(2-3):169-95.

Koles ZJ. Trends in EEG source localization. Electroencephalography and clinical Neurophysiology. 1998 Feb 1;106(2):127-37.

Kramer D, Bommer PL, Tombolini C, Koppe G, Durstewitz D. Reconstructing nonlinear dynamical systems from multi-modal time series. arXiv preprint arXiv:2111.02922. 2021 Nov 4.

Kucukelbir A, Tran D, Ranganath R, Gelman A, Blei DM. Automatic differentiation variational inference. Journal of machine learning research. 2017.

La Vaque TJ. The history of EEG Hans Berger: psychophysiologist. A historical vignette. Journal of Neurotherapy. 1999 Apr 1;3(2):1-9.

Lai HY, Albaugh DL, Kao YC, Younce JR, Shih YY. Robust deep brain stimulation functional MRI procedures in rats and mice using an MR‐compatible tungsten microwire electrode. Magnetic resonance in medicine. 2015 Mar;73(3):1246-51.

Lam SK, Pitrou A, Seibert S. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC 2015 Nov 15 (pp. 1-6).

Laufs H, Kleinschmidt A, Beyerle A, Eger E, Salek-Haddadi A, Preibisch C, Krakow K. EEG-correlated fMRI of human alpha activity. Neuroimage. 2003 Aug 1;19(4):1463-76.

Lechner M, Hasani R, Amini A, Henzinger TA, Rus D, Grosu R. Neural circuit policies enabling auditable autonomy. Nature Machine Intelligence. 2020 Oct;2(10):642-52.

Lee C. 18. Itō Calculus [Internet]. MIT 18.S096 Topics in Mathematics with Applications in Finance, Fall 2013 [Internet]. 2013 [cited 2023 Oct 19]. Available from: https://www.youtube.com/watch?v=Z5yRMMVUC5w&t=1991s&ab_channel=MITOpenCourseWare

Li X, Wong TK, Chen RT, Duvenaud D. Scalable gradients for stochastic differential equations. In International Conference on Artificial Intelligence and Statistics 2020 Jun 3 (pp. 3870-3882). PMLR.

Liu J, Lee HJ, Weitz AJ, Fang Z, Lin P, Choy M, Fisher R, Pinskiy V, Tolpygo A, Mitra P, Schiff N. Frequency-selective control of cortical and subcortical networks by central thalamus. Elife. 2015;4.

Sanz Leon P, Knock SA, Woodman MM, Domide L, Mersmann J, McIntosh AR, Jirsa V. The Virtual Brain: a simulator of primate brain network dynamics. Frontiers in neuroinformatics. 2013 Jun 11;7:10.

Schirner M, Deco G, Ritter P. Learning how network structure shapes decision-making for bio-inspired computing. Nature Communications. 2023 May 23;14(1):2963.

Maruyama G. Continuous Markov processes and stochastic equations. Rendiconti del Circolo Matematico di Palermo. 1955 Jan;4:48-90.

McBride L, Narendra K. Optimization of time-varying systems. IEEE Transactions on Automatic Control. 1965 Jul;10(3):289-94.

Michel CM, Brunet D. EEG source imaging: a practical review of the analysis steps. Frontiers in neurology. 2019 Apr 4;10:325.

Momi D, Wang Z, Griffiths JD. TMS-evoked responses are driven by recurrent large-scale network dynamics. Elife. 2023;12.

Montes-Restrepo V, van Mierlo P, Strobbe G, Staelens S, Vandenberghe S, Hallez H. Influence of skull modeling approaches on EEG source localization. Brain topography. 2014 Jan;27:95-111.

Moosmann M, Ritter P, Krastel I, Brink A, Thees S, Blankenburg F, Taskin B, Obrig H, Villringer A. Correlates of alpha rhythm in functional magnetic resonance imaging and near infrared spectroscopy. Neuroimage. 2003 Sep 1;20(1):145-58.

Moran R, Pinotsis DA, Friston K. Neural masses and fields in dynamic causal modeling. Frontiers in computational neuroscience. 2013 May 28;7:57.

Negahbani E, Steyn-Ross DA, Steyn-Ross ML, Wilson MT, Sleigh JW. Noise-induced precursors of state transitions in the stochastic Wilson–Cowan model. The Journal of Mathematical Neuroscience (JMN). 2015 Dec;5:1-27.

Pang JC, Robinson PA. Neural mechanisms of the EEG alpha-BOLD anticorrelation. Neuroimage. 2018 Nov 1;181:461-70.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems. 2019;32.

Pearlmutter BA. Gradient calculations for dynamic recurrent neural networks: A survey. IEEE Transactions on Neural networks. 1995 Sep;6(5):1212-28.

Pineda FJ. Generalization of back-propagation to recurrent neural networks. Physical review letters. 1987 Nov 9;59(19):2229.

Poli M, Massaroli S, Yamashita A, Asama H, Park J, Ermon S. TorchDyn: implicit models and neural numerical methods in PyTorch. In Neural Information Processing Systems, Workshop on Physical Reasoning and Inductive Biases for the Real World 2021 (Vol. 2).

Ritter P, Villringer A. simultaneous EEG–fMRI. Neuroscience & Biobehavioral Reviews. 2006 Jan 1;30(6):823-38.

Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. nature. 1986 Oct 9;323(6088):533-6.

Schirner M, McIntosh AR, Jirsa V, Deco G, Ritter P. Inferring multi-scale neural mechanisms with brain network modelling. Elife. 2018 Jan 8;7:e28927.

Schirner M, Ritter P. Integrating EEG–fMRI Through Brain Simulation. InEEG-fMRI: Physiological Basis, Technique, and Applications 2023 Jan 1 (pp. 745-777). Cham: Springer International Publishing.

Strang G. Differential equations and linear algebra. Wellesley, Ma: Wellesley-Cambridge Press; 2014.

Sherstinsky A. Deriving the recurrent neural network definition and rnn unrolling using signal processing. In Critiquing and Correcting Trends in Machine Learning Workshop at Neural Information Processing Systems 2018 Dec (Vol. 31, p. 4).

Wallace E, Benayoun M, Van Drongelen W, Cowan JD. Emergent oscillations in networks of stochastic spiking neurons. Plos one. 2011 May 6;6(5):e14804.

Wang XJ. Probabilistic decision making by slow reverberation in cortical circuits. Neuron. 2002 Dec 5;36(5):955-68.

Werbos PJ. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE. 1990 Oct;78(10):1550-60.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M. Welcome to the Tidyverse. Journal of open source software. 2019 Nov 21;4(43):1686.

Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. Neural computation. 1989 Jun 1;1(2):270-80.

Williams RJ, Peng J. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. Neural computation. 1990 Dec 1;2(4):490-501.

Wilson FA, O'scalaidhe SP, Goldman-Rakic PS. Functional synergism between putative gamma-aminobutyrate-containing neurons and pyramidal neurons in prefrontal cortex. Proceedings of the National Academy of Sciences. 1994 Apr 26;91(9):4009-13.

Wilson HR, Cowan JD. Excitatory and inhibitory interactions in localized populations of model neurons. Biophysical journal. 1972 Jan 1;12(1):1-24.

Wilson HR, Cowan JD. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. Kybernetik. 1973 Sep;13(2):55-80.

Wong KF, Wang XJ. A recurrent network mechanism of time integration in perceptual decisions. Journal of Neuroscience. 2006 Jan 25;26(4):1314-28.

Zhang F, Daducci A, He Y, Schiavi S, Seguin C, Smith RE, Yeh CH, Zhao T, O'Donnell LJ. Quantitative mapping of the brain's structural connectivity using diffusion MRI tractography: A review. Neuroimage. 2022 Apr 1;249:118870.