

# Unravelling Brain Networks in Chronic Pain and Spinal Cord Stimulation through Magnetoencephalography and Graph Neural Networks



Master Thesis Technical Medicine

Caroline Witstok

May 2025

This page was left blank intentionally.

# Unravelling Brain Networks in Chronic Pain and Spinal Cord Stimulation through Magnetoencephalography and Graph Neural Networks

Caroline Witstok

Student number: 4883586

23 May 2025

Thesis in partial fulfilment of the requirements for the joint degree of Master of Science in

## *Technical Medicine*

Leiden University; Delft University of Technology; Erasmus University Rotterdam

Master thesis project (TM30004; 35 ECTS)

Department of Pain Medicine, Erasmus Medical Center

October 2024 - May 2025

Supervisors:

Dr. ir. Cecile de Vos

Dr. ir. Mark van de Ruit

Dr. Sander Frankema

Drs. Laurien Reinders

Thesis committee members:

Dr. ir. Cecile de Vos (chair)

Dr. ir. Mark van de Ruit

Dr. Robert van den Berg

Dr. ir. Arjan Hillebrand

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



This page was left blank intentionally.

# | Summary

**Introduction:** Chronic pain is a widespread and complex condition. Spinal Cord Stimulation (SCS) offers effective pain relief in a portion of patients suffering from chronic pain, although its underlying mechanisms of action remain unclear and may differ between tonic and burst stimulation paradigms. Brain connectivity analysis can help reveal how chronic pain and SCS affect communication between brain regions. Magnetoencephalography (MEG) is particularly suited for this due to its high temporal resolution. Graph theory enables modelling of whole-brain networks, and Graph Neural Networks (GNNs), a deep learning approach designed for graph-structured data, is well-suited for distinguishing specific connectivity patterns within complex network structures. While promising, GNNs have not yet been applied to SCS or chronic pain. Furthermore, beyond classification, explainability approaches allow insights into which graph substructures drive GNN model's decisions.

**Aim:** The overarching aim of my exploratory study was to develop and train a GNN model based on MEG data from patients with chronic pain with SCS, to identify differences in brain networks during stimulation ON and OFF.

**Methods:** Resting-state MEG data were collected from 22 chronic pain patients receiving SCS, recorded in two institutes. A cyclic stimulation protocol (1 min ON, 1 min OFF) was used. Brain connectivity graphs were constructed using the phase lag index as functional connectivity metric, and features for each brain region were derived from the power spectral density. Graph datasets were created per frequency band, stimulation paradigm (tonic and burst), and recording institute. Separate GNN models were trained to classify stimulation ON and OFF states, and explainability techniques were implemented to unravel the key graph substructures driving the model's classification decisions.

**Results:** GNN models accurately classified stimulation states, especially using full-band, beta, and gamma graphs (accuracies: 0.99, 0.97, 0.99). Delta, theta, and alpha bands showed lower performance (accuracies: 0.76, 0.80, 0.77). Model performance was consistent across tonic and burst SCS paradigms and both recording sites (accuracies: 0.97, 0.98, 0.99, 0.97), however, performance across paradigms showed inconsistencies. Specifically, the model trained on tonic SCS and tested on burst SCS recordings showed a cross-paradigm accuracy of only 0.69. The GNN model achieved cross-site accuracies of 0.81 and 0.87 across datasets from the recording institutes, demonstrating consistent performance across patient cohorts. Furthermore, the explainability analysis outcomes highlighted several pain-related brain regions as key substructures in the graph for distinguishing stimulation ON and OFF states.

**Discussion:** This study introduces GNNs as a novel method for decoding brain network dynamics in chronic pain patients with SCS. The classification results and node-level explainability suggest that pain-processing regions are modulated by SCS. The cross-paradigm accuracy suggests that burst SCS only partially captures the features of tonic SCS, possibly indicating that burst SCS engages a more widespread brain network. However, interpretation of the findings is limited by the small sample size, inter-patient variability, and the inability to separate chronic pain effects from stimulation effects. Nevertheless, this framework offers a promising direction for application of GNNs for unravelling complex brain network dynamics in chronic pain and SCS. Future studies should focus on expanding this framework by utilizing GNN models to classify SCS treatment effectiveness, potentially providing more insights into the brain regions and connectivity patterns that are most predictive of treatment success.

# | Preface

This thesis marks the conclusion of my graduation project and, with this, my time as a student at Delft University of Technology, Erasmus University Rotterdam, and Leiden University almost comes to an end. During my bachelor's in Clinical Technology, I became increasingly fascinated by this relatively new and evolving field of technological advancements in healthcare. Following my bachelor, I enrolled in the master's program in Technical Medicine, specializing in the Sensing and Stimulation track. During the technical courses, my interest in signal analysis and programming was further sparked. After completing the first year of courses, I sought practical experience and took some time off from my studies to work at Quantib, a MedTech company, where I was part of the research and development team. Contributing clinical insights to the development of radiology software gave me valuable industry experience and broadened my understanding of the role of a Technical Physician. This experience reinforced my motivation to continue the master's program.

For my master thesis, I was fortunate to work on a project that integrated signal analysis, programming, and neuroscience, fields that greatly fascinate me. This research allowed me to explore the chronic pain patient population, deepen my understanding of brain function, and gain knowledge in a novel research area. It challenged me to continuously consider the clinical relevance of this research and to make the findings interpretable for clinical applications. Additionally, I had the opportunity to supervise students during their internship projects, which further enriched my academic experience. I am excited to continue this research in the coming months as part of my electives period, during which I hope to further expand my knowledge and expertise.

I would like to express my sincere gratitude to everyone who supported me during my graduation project. Cecile, thank you for your guidance, collaboration, and the many insightful conversations—both about this project and beyond. Your enthusiasm inspired and motivated me to explore new perspectives in my research. Laurien, I am very grateful for your mentorship, expertise, and encouragement throughout the project. Mark, your feedback on the technical aspects of my research was incredibly valuable and helped sharpen my analytical thinking. Arjan and Robert, thank you for your involvement as committee members and for sharing your expertise. I also want to thank Julia, Dido, and Cathelijne for your contributions to the project, your useful insights, and for making the work in this research area feel more collaborative. To my friends, housemates, and everyone in Na-17, thank you for your support and the joy you brought outside of my thesis. Lastly, I want to give special thanks to my family and to Jort for your unconditional support, whether through insightful discussions that brought me new perspectives, celebrating breakthrough moments, or offering constant encouragement.

*Caroline Witstok  
Delft, May 2025*

# Contents

<b>Summary</b>	<b>ii</b>
<b>Preface</b>	<b>iii</b>
<b>Nomenclature</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Chronic pain . . . . .	1
1.2 Spinal Cord Stimulation . . . . .	1
1.3 Brain Connectivity . . . . .	2
1.4 Graph Theory . . . . .	3
1.5 Graph Neural Networks . . . . .	4
1.6 Aim . . . . .	6
<b>2 Methods</b>	<b>7</b>
2.1 Data Acquisition . . . . .	7
2.1.1 Patients . . . . .	7
2.1.2 Study Protocol . . . . .	7
2.1.3 Recordings . . . . .	8
2.2 Data Processing . . . . .	8
2.2.1 Software . . . . .	8
2.2.2 Data Cleaning . . . . .	8
2.2.3 Sensor Space and Source Space Analysis . . . . .	9
2.2.4 Epoching . . . . .	10
2.3 Graph Dataset . . . . .	11
2.3.1 Node Selection . . . . .	11
2.3.2 Node Feature Computation . . . . .	12
2.3.3 Edge Definition and Edge Weight Computation . . . . .	13
2.4 Graph Neural Network Model . . . . .	14
2.4.1 Model Architecture . . . . .	14
2.4.2 Model Training . . . . .	17
2.4.3 Hyperparameter Optimization . . . . .	18
2.4.4 Model Evaluation . . . . .	21
2.4.5 Node Explainability . . . . .	23
<b>3 Results</b>	<b>25</b>
3.1 Patient Population . . . . .	25
3.2 Graphs . . . . .	25
3.2.1 Sensor Space Analysis . . . . .	25
3.2.2 Source Space Analysis . . . . .	27
3.3 Model Outcomes . . . . .	29
3.3.1 Model Performance . . . . .	29
3.3.2 Receiver Operating Characteristic Curves . . . . .	31

3.3.3	Hyperparameter Settings . . . . .	33
3.3.4	Training and Validation Accuracy and Loss . . . . .	33
3.4	Node Explainability . . . . .	34
3.4.1	Sensor Space . . . . .	35
3.4.2	Source Space . . . . .	36
<b>4</b>	<b>Discussion</b>	<b>38</b>
4.1	Graph Dataset . . . . .	38
4.1.1	Node Features . . . . .	38
4.1.2	Edges . . . . .	39
4.1.3	Frequency Band Graphs . . . . .	40
4.1.4	Class Imbalance . . . . .	40
4.2	Graph Neural Network Model . . . . .	41
4.3	Model Performance . . . . .	41
4.3.1	Frequency Bands . . . . .	41
4.3.2	Stimulation Paradigms . . . . .	42
4.3.3	Recording Sites . . . . .	42
4.3.4	Source Space Analysis . . . . .	42
4.4	Model Training . . . . .	43
4.4.1	Model Hyperparameters . . . . .	43
4.4.2	Model Training . . . . .	44
4.5	Node Explainability . . . . .	44
4.5.1	Important Nodes . . . . .	44
4.5.2	Interpretation . . . . .	45
4.6	General Strengths and Limitations . . . . .	46
4.6.1	Analysis Framework . . . . .	46
4.6.2	Design Choices . . . . .	47
4.6.3	Dataset . . . . .	47
4.6.4	Computational Demands . . . . .	48
4.6.5	Carry-over Effects . . . . .	49
4.7	Recommendations . . . . .	49
<b>5</b>	<b>Conclusion</b>	<b>51</b>
	<b>References</b>	<b>52</b>
	<b>Appendices</b>	<b>61</b>
<b>A</b>	<b>Data Processing Pipeline</b>	<b>62</b>
<b>B</b>	<b>Brainstorm and Python Integration</b>	<b>63</b>
B.1	Sensor Space Data . . . . .	63
B.2	Source Space Data . . . . .	63
<b>C</b>	<b>Preprocessing Steps</b>	<b>64</b>
C.1	Data Cleaning . . . . .	64
C.1.1	Notch Filters . . . . .	64
C.1.2	Bad Channel Removal . . . . .	64
C.1.3	Additional Artifact Removal . . . . .	65
C.1.4	Exclusion of Bad Segments . . . . .	65

---

C.2	Head Model and Source Estimation . . . . .	65
C.2.1	Anatomy . . . . .	65
C.2.2	Source Space . . . . .	66
C.2.3	Surface Model . . . . .	66
C.2.4	Source Estimation . . . . .	66
C.2.5	Scout Time Series . . . . .	66
<b>D</b>	<b>Regions of Interest</b>	<b>67</b>
D.1	Scout Acronyms . . . . .	69
<b>E</b>	<b>Subepoch Duration and Overlap</b>	<b>70</b>
<b>F</b>	<b>Bad Subepoch Detection</b>	<b>72</b>
<b>G</b>	<b>MEG Channel Selection</b>	<b>73</b>
<b>H</b>	<b>Node Feature Normalization</b>	<b>75</b>
<b>I</b>	<b>Phase Lag Index Computation</b>	<b>77</b>
<b>J</b>	<b>Node Explainability using SubgraphX</b>	<b>78</b>
J.1	SubgraphX explainer . . . . .	78
J.1.1	Monte Carlo Tree Search . . . . .	78
J.1.2	Shapley Values . . . . .	78
J.2	Fidelity and Sparsity . . . . .	79
J.3	Parameters . . . . .	80
<b>K</b>	<b>Patient Population</b>	<b>81</b>
K.1	Patient Characteristics . . . . .	81
K.2	Stimulation Parameters . . . . .	82
<b>L</b>	<b>Connectivity Differences and Node Features</b>	<b>83</b>
<b>M</b>	<b>Accuracy and Loss curves</b>	<b>89</b>
<b>N</b>	<b>Confusion Matrices</b>	<b>93</b>

# Nomenclature

---

<b>Notation</b>	<b>Description</b>
ACC	anterior cingulate cortex
AI	Artificial Intelligence
AUC	Area Under the Curve
CRPS	Complex Regional Pain Syndrome
DC	Direct Current
DLPFC	dorsolateral prefrontal cortex
EEG	Electroencephalography
FIR	Finite Impulse Response
fMRI	functional Magnetic Resonance Imaging
FN	False Negatives
FP	False Positives
GNN	Graph Neural Network
IIR	Infinite Impulse Response
M1	primary motor cortex
MAD	Median Absolute Difference
MCC	middle cingulate cortex
MCTS	Monte Carlo Tree Search
MEG	Magnetoencephalography
MN	Minimum Norm
MNI	Montreal Neurological Institute
MRI	Magnetic Resonance Imaging
NRS	Numeric Rating Scale
OFC	orbitofrontal cortex
PCA	Principle Component Analysis
PCC	posterior cingulate cortex
PLI	Phase Lag Index
PSD	Power Spectral Density
PSPS-2	Persistent Spinal Pain Syndrome type 2

---

---

<b>Notation</b>	<b>Description</b>
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
S1	primary somatosensory cortex
S2	secondary somatosensory cortex
SCS	Spinal Cord Stimulation
SMA	supplementary motor area
TN	True Negatives
TP	True Positives

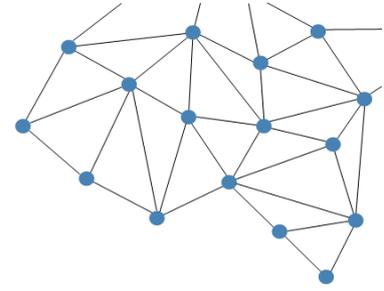
---

## Symbols

---

<b>Description</b>	<b>Unit</b>	<b>Symbol</b>
Frequency	Hertz	[Hz]
Normalized Node Feature	Arbitrary Unit	[AU]
Signal Ratio	Decibel	[dB]

---



# 1 | Introduction

## 1.1. Chronic pain

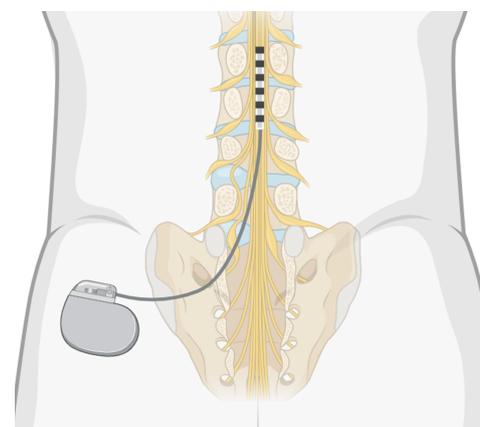
Chronic pain is a pervasive and debilitating condition that affects approximately 20% of the world population [1]. Chronic pain is defined by the International Association for the Study of Pain as pain that persists or recurs for more than three months, that is significantly interfering with daily functioning and often is accompanied by distress [2]. The burden of chronic pain is multifaceted, encompassing health care costs, lost productivity, and the emotional and psychological toll on individuals. Despite its widespread impact, chronic pain remains a complex condition to diagnose and treat, largely because pain is inherently subjective and influenced by emotional, cognitive, and environmental factors, and presents with large variability in physiological aspects. Furthermore, the underlying brain mechanisms that cause and maintain chronic pain are still not fully understood. This complicates accurate quantification and characterization, thereby impeding advancements in the understanding of chronic pain and adequate treatment.

## 1.2. Spinal Cord Stimulation

Spinal Cord Stimulation (SCS) is an advanced neuromodulation treatment option for patients suffering from chronic pain that have exhausted all other options. SCS involves surgically implanting a system with one or two electrode leads placed in the epidural space connected to an implanted pulse generator, which can deliver electrical pulses to the dorsal columns of the spinal cord to achieve pain reduction. The analgesic effects of SCS appear to involve both spinal and supraspinal mechanisms [3, 4]. A schematic overview of SCS is illustrated in Figure 1.1.

The stimulation can be customized with different parameters, such as amplitude and frequency, offering various stimulation paradigms, including tonic and burst modes. Tonic stimulation delivers pulses with a consistent frequency, pulse width, and amplitude to the dorsal columns, typically producing paresthesias in the regions innervated by the targeted nerve fibres, which creates a feeling of numbness, burning, or tingling [5, 6]. More recently, paresthesia-free paradigms have been developed, such as burst SCS [7]. This approach uses bursts of five high-frequency pulses delivered at lower amplitudes, mimicking natural thalamocortical firing patterns [8]. Evidence suggests that burst SCS may act through distinct mechanisms compared to tonic stimulation [9–11]. While tonic SCS is thought to influence both the ascending lateral pain pathway and the descending pain inhibitory system, burst SCS may also engage the ascending medial pain pathway [12–14].

Depending on the pain aetiology, this therapy provides over 50% pain reduction in approximately 65%



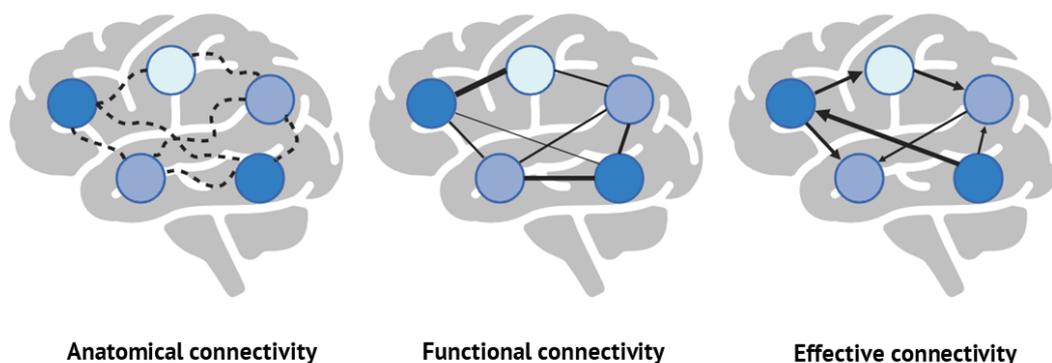
**Figure 1.1:** Schematic overview of Spinal Cord Stimulation, with one electrode in the epidural space, connected to an implanted pulse generator.

of the patients [15–17]. However, since the brain mechanisms underlying chronic pain and SCS effects remain poorly understood, it is challenging to predict treatment success, and additionally, it hinders the personalization and optimization of chronic pain management.

### 1.3. Brain Connectivity

In recent years, brain connectivity analysis has been increasingly important within the field of neuroscience to gain insights into both anatomical and functional organization of the brain [18, 19]. Brain connectivity refers to the complex network of neural connections within the brain, which are crucial for the brain's ability to process information, perform cognitive functions, and regulate behaviour [20–22]. By mapping and analysing these connections, a comprehensive view of the brain as a complex network and its interactions can be acquired. Brain connectivity analysis is valuable for understanding neural mechanisms underlying chronic pain. Unlike neurological diseases where the primary pathology is in the brain, chronic pain involves peripheral injury or pathology that is thought to lead to an imbalance between pain input and pain suppression, affecting pain perception and processing [23]. This imbalance can be analysed with brain connectivity, which has the potential to broaden the understanding of chronic pain as well as the working mechanisms of treatment, such as SCS.

Brain connectivity can be broadly categorized into three types: anatomical, functional, and effective connectivity. The different types of connectivity are illustrated in Figure 1.2. Anatomical connectivity describes the physical wiring of the brain, encompassing the anatomical connections between neurons and brain regions. Functional and effective connectivity, on the other hand, involve dynamic interactions between brain regions. Functional connectivity refers to the temporal correlation and synchronization of neural activity between spatially distant brain regions. Effective connectivity captures the directionality and causal relationship between brain regions, aiming to understand how activity in one area of the brain affects activity in another area [24–26].



**Figure 1.2:** Anatomical, functional, and effective connectivity.

Several studies have employed different connectivity metrics to analyse functional interactions using neuroimaging data [27–29]. The calculations of these metrics are highly dependent on capturing the temporal dynamics of neural interactions between different brain regions, making high spatial and temporal resolution techniques essential for accurate analysis. Therefore, modalities offering millisecond-level temporal resolution are particularly suitable for calculating connectivity metrics [28]. Magnetoencephalography (MEG) captures the oscillatory brain activity in real time with high temporal resolution by detecting changes in the magnetic field related to neuronal activity in the brain.

While brain connectivity measures offer valuable insights into the relationships and interactions between specific pairs of brain regions, they lack the ability to capture the complex network of the brain. Single

connectivity measures only consider pairwise interactions, however, the brain operates as a highly interconnected network where interactions are not isolated but part of a dynamic and integrative network [30]. Similarly, chronic pain is driven by complex mechanisms involving multiple neural pathways across multiple brain areas, including both ascending nociceptive pathways that transmit pain signals and descending pathways that modulate these signals. This multifaceted nature underscores the complex network of neural activity in chronic pain.

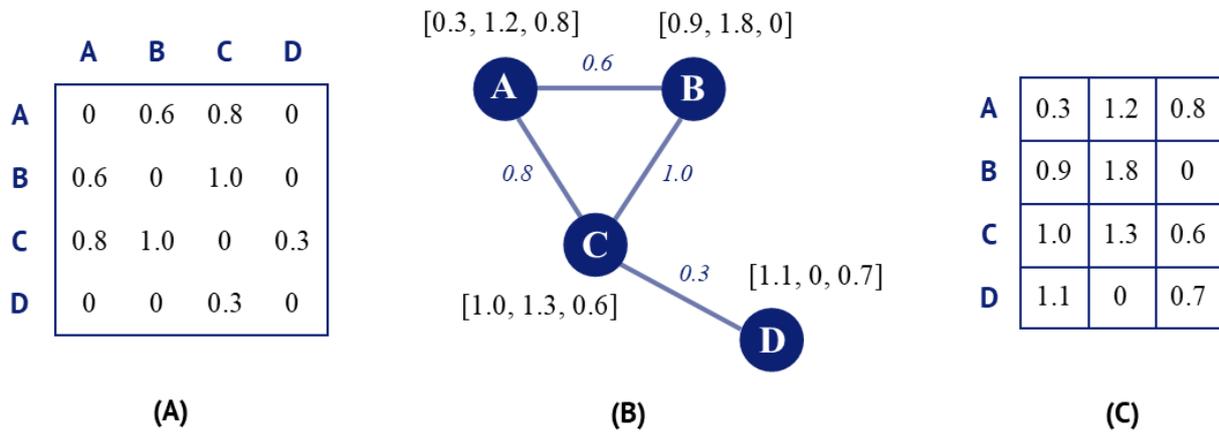
Additionally, mechanisms of action of SCS are thought to be based on changing the neural activity in several spinal and supraspinal pathways, potentially restoring normal connections [31]. Previous studies using functional Magnetic Resonance Imaging (fMRI) suggest that SCS may reduce the strength of connections between pain-processing regions in the brain [32]. However, the brain connectivity across various active networks in chronic pain and in SCS remains underexplored.

## 1.4. Graph Theory

Analysing brain connectivity using graph theory has been shown to be a powerful approach for analysing brain networks, due to its ability to depict the functional structure of the brain as an interconnected complex network [33, 34]. Graph theory offers a mathematical framework for complex network analysis by representing data as complex networks, referred to as graphs. Graphs are abstract representations of relationships between objects, facilitating the modelling of the brain as a complex network. This approach captures not only individual connections but also the emergent structure and organization of the entire brain network. Graph theory methods are increasingly applied in neuroscience to account for the complex, high-dimensional interactions between brain regions [35]. Furthermore, graph theory provides meaningful insights into the organization of human brain networks by quantifying how brain regions are connected at both global and local levels [36]. By analysing graphs at a global level, the overall structures and properties of the entire brain network can be researched, providing information on the connectedness and efficiency of the network. Contrarily, the local level focusses on the properties and connectivity of individual nodes or small groups of nodes within the brain network. At this level, interactions within specific brain regions can be examined.

A graph ( $G$ ) is a collection of nodes ( $N$ ) interconnected by edges ( $E$ ) and a node feature matrix ( $X$ ), and can be written as  $G = \{N, E, X\}$ . Graph edges can be weighted or unweighted, as well as directed or undirected. Weighted edges provide information about the strength of the connections, while unweighted or binary edges only indicate the presence or absence of connections. Directed graphs represent information flow in one direction from one node to another node. On the other hand, in an undirected graph, every edge represents a two-way connection between nodes, so information can flow in both directions along any edge [37, 38]. The set of edges  $E$  is given by the adjacency matrix, which is a square  $N \times N$  matrix representing the graph and its connectivity structure, including the weights of each connection for a weighted graph. The node feature matrix is an  $N \times C$  matrix, where each row is a feature representation of a node consisting of a  $[1, C]$  feature vector. The node features provide information on the node.

Graph theory can be utilized to model the anatomical, functional, and effective connections in complex brain networks. When using graph theory to model brain connectivity, nodes commonly represent brain regions, and edges represent a measure of connectivity between the nodes, consisting of anatomical, functional, or effective connections [35]. Node features can be structural information of the node's role within a graph, such as the degree of the node (the number of connections), or signal-based information that can be derived from the data, for instance a statistical or frequency-based measure. A visualization of a weighted undirected graph, with the corresponding adjacency matrix, the graph structures (nodes, edges, and node features), and the node feature matrix can be seen in Figure 1.3. For simplification, the



**Figure 1.3:** Visualization of a graph's adjacency matrix (A), the graph structure including the nodes, edges, and node features (B), and the node feature matrix (C).

feature representations of the nodes are composed of three features.

This mathematical network representing brain activity and connections between brain areas is referred to as connectomics [33, 34]. Using connectomics, the interdependencies and structures of intricate brain networks can be inferred [30]. By analysing connectomics in chronic pain patients, patterns of connectivity and their relationship to chronic pain can be explored, as well as effects on connectivity through SCS therapy. Previous research on chronic pain through graph theory has revealed distinctive brain network alterations that are crucial for understanding the condition [39, 40]. These changes, which predominantly occur at a local level with some influence on the global network, indicate that specific brain regions undergo significant shifts in their roles within the broader network. Identifying these network alterations is beneficial as it provides insights into the clinical manifestations of chronic pain, including correlations with pain intensity and duration, and can lead to the development of personalized treatment strategies.

## 1.5. Graph Neural Networks

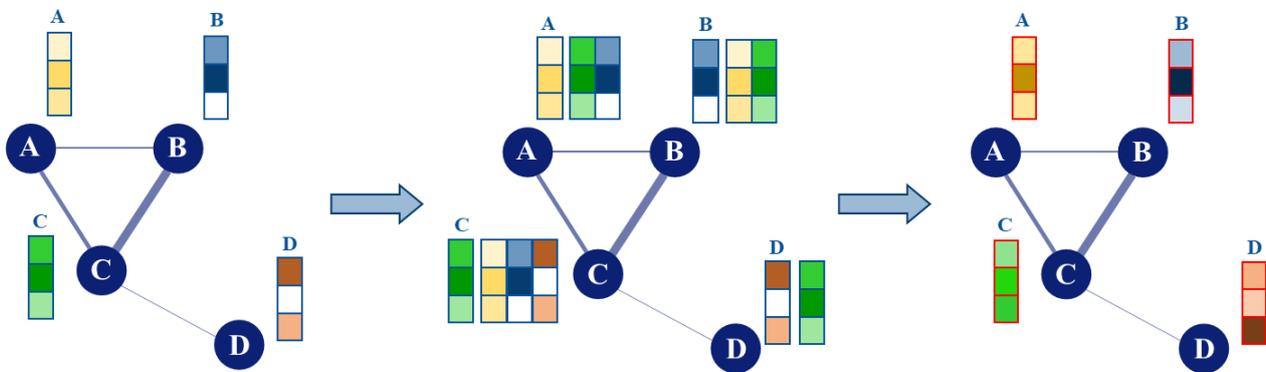
The integration of Artificial Intelligence (AI) for enhancing the accuracy and efficiency of diagnostic processes is rapidly evolving. AI is a broad field encompassing machine learning and deep learning methods, where machine learning refers to algorithms that learn patterns from data, and deep learning is a specialized subset that uses neural networks to automatically learn features from large and complex datasets. AI is particularly effective in analysing complex data, such as Electroencephalography (EEG) or MEG signals, and is thought to be well-suited for analysing brain connectivity on a network scale due to its ability to handle complex and high-dimensional data [41, 42].

Traditional machine learning methods require domain knowledge and heavily rely on manual feature extraction [43–45]. Even though these approaches focus on known clinically relevant features and thereby offer interpretability of the model, this manual feature selection process can introduce subjectivity and biases and is cumbersome. Deep learning methods, such as Convolutional Neural Networks and Long Short-Term Memory networks, have shown promise by automatically extracting spatial and temporal features in EEG signals [46–48]. Nevertheless, classical deep learning algorithms are not designed to efficiently analyse brain connectivity networks, since they operate on regular Euclidean data like texts, images, and single time-series data [49].

Extending the application of deep learning on non-Euclidean domains that encompass graphs has been an emerging research area [50–52]. Recent advancements have led to the development of Graph Neural Networks (GNNs), which are deep learning methods designed to operate on graph-structured data. GNNs

were introduced in Gori et al. [53] and Scarselli et al. [54] as a generalization of neural networks that can directly operate on graphs. In a GNN model, graphs are used as input and an iterative process of aggregating and updating information from the graphs enables learning intricate patterns in complex network data [54, 55]. GNNs uniquely excel in modelling complex, multi-channel time series data and the interactions between these channels, offering advantages over traditional deep learning methods [56]. Even though the construction of the input graphs requires a manual step that can heavily influence model performance, GNNs have the advantage of being able to learn higher-order feature representations beyond the initially defined inputs, potentially capturing complex relational patterns in the data that are not explicitly engineered beforehand.

The input of a GNN is a graph in the form  $G = \{N, E, X\}$ , where each node ( $N$ ) in the graph has an initial representation consisting of the node features ( $X$ ) and the edges between each node are included in a connectivity matrix ( $E$ ). By utilizing a convolution mechanism, GNN models facilitate the exchange of information between neighbouring nodes, including the neighbouring node features and edges, enabling the model to learn intricate patterns within the graph [57]. The updated node feature representations are often referred to as updated node embeddings, containing enriched information from the graph context. This process is incorporated in convolutional layers, which are the main part of a GNN model. This convolution mechanism is illustrated in Figure 1.4. This iterative updating and processing of information makes GNNs particularly suitable for brain connectivity analysis, where intricate patterns in brain networks can be uncovered. These patterns can provide valuable insights into cognitive processes, neurological disorders, and therapeutic effects, and hence, GNNs have been applied successfully for research on neurodegenerative diseases [56, 58, 59]. Following the convolutional layers, the GNN uses the updated node embeddings to learn distinguishing patterns between the input classes, ultimately enabling the model to perform classification.



**Figure 1.4:** Illustration of the convolution mechanism inside a GNN model, in which the initial node features (represented as colours in the left graph) are updated. The node features of the neighbouring nodes are aggregated (middle graph) and weighted based on the edge weight between this node and its neighbouring node, illustrated as line thickness. Finally, updated node embeddings are acquired for each node (right graph).

In the context of chronic pain and SCS, GNNs have the possibility to reveal how pain alters brain connectivity, and which brain areas are mostly involved in pain processing and modulation through therapy. Identification of specific network disruptions associated with pain processing and therapeutic effects can lead to a better understanding of chronic pain and SCS. The application of GNNs on chronic pain research has not been explored yet, but provides a promising new approach, as GNNs can leverage the complex patterns within the connectome to distinguish between healthy individuals, chronic pain patients, and the efficacy of SCS treatments. Thereby, GNN models have the potential to effectively distinguish specific connectivity patterns within the complex connectivity network, which may provide new insights into the

underlying mechanisms of chronic pain and SCS.

Drawing from my review on GNN-based approaches for investigating brain connectivity as complex networks, there is a growing interest in leveraging GNNs to unravel intricate brain network patterns. These models offer a powerful tool for identifying connectivity dynamics that traditional methods may overlook. However, their application to chronic pain and SCS remains largely unexplored [60]. My literature review also highlighted significant disparities in analytical approaches across different GNN applications and provides recommendations for a GNN framework focused on chronic pain research. Given the novelty of this approach, there is currently no clear consensus on the optimal implementation of GNNs in this domain. Moreover, similar to other deep learning techniques, GNNs inherent "black box" nature complicates the understanding of the model's classification output, making it crucial to develop strategies that enhance their interpretability.

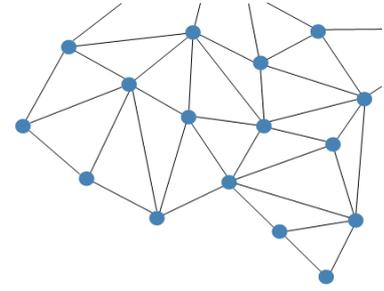
To address this, explainability methods can be employed to uncover what features the model bases its decisions on and which network components are most influential in the classification process. Various explainability techniques have been proposed for GNN-based models, but their application in neuroscience remains underdeveloped, with limited published work and scarce availability of source code [61–64]. The integration of an explainability framework can facilitate the identification of the active brain networks associated with the effect of SCS in chronic pain. This approach bridges the gap between the abstract graph-based representations used in the GNN model and their clinical relevance. By identifying the most relevant brain regions and connectivity patterns for the classification outcomes, this framework enhances both the scientific understanding of pain-related brain network dynamics and the potential for clinically meaningful applications of GNN-based models in chronic pain research.

## 1.6. Aim

The overarching aim of this exploratory study was to develop and train a Graph Neural Network (GNN) model based on Magnetoencephalography (MEG) data from patients with chronic pain treated with Spinal Cord Stimulation (SCS), with the ultimate goal of identifying differences in active brain networks during stimulation on and off.

To achieve this, this study pursued the following objectives:

1. **Preprocessing and transforming MEG recordings into graph-structured datasets:** This objective focused on preprocessing MEG data, defining the graph structures, and extracting the necessary features to construct graph-based representations.
2. **Designing and implementing a GNN model to classify brain networks associated with SCS effects in chronic pain patients:** A GNN model architecture was designed and developed with the goal of learning and distinguishing the brain network patterns associated with SCS.
3. **Evaluating model performance across multiple graph input definitions:** Multiple graph datasets were generated with different graph structure definitions. Separate models were trained and compared to evaluate which graph-based representation best captured the effects of SCS.
4. **Applying explainability methods to interpret model decisions and uncover relevant brain regions:** This objective is implemented to uncover which substructures in the input graphs were most influential in the model's classification decisions. This step was critical to improve model interpretability and to allow identification of brain regions potentially involved in chronic pain and modulation through SCS.



## 2 | Methods

### 2.1. Data Acquisition

In this study, MEG data acquired from two institutions was utilized: the Montreal Neurological Institute (MNI) at McGill University in Canada and the Donders Institute of Radboud University in Nijmegen, the Netherlands. Data was collected by Bart Witjes and Cecile de Vos, researchers at the Erasmus Medical Center. The recordings took place between July 2018 and May 2019. The study was approved by the Institutional Review Board of the Montreal Neurological Institute and the Committee for Human-Related Research (Dutch: Commissie Mensgebonden Onderzoek) region Arnhem-Nijmegen. All patients included in the study provided written informed consent.

#### 2.1.1. Patients

The data set included recordings from a total of 22 chronic pain patients treated with SCS, with 10 patients recorded in Montreal and 12 in Nijmegen. All patients reported chronic pain in their lower back and/or lower extremity. An overview of the patient population and characteristics is provided in Table K.1.

#### 2.1.2. Study Protocol

At the start of each measurement session, the current pain score attributed to the chronic pain condition was assessed using the Numeric Rating Scale (NRS) from zero to ten, where zero represents no pain and ten represents the worst pain imaginable. The patients underwent three measuring sessions, during each of which they were treated with a different SCS paradigm: tonic SCS, burst SCS, or a sham SCS paradigm.

For this thesis, MEG recordings of resting-state neuronal activity measured during tonic and burst SCS were used. The resting-state recordings varied in length from six to ten minutes. During the recording session, patients were instructed to remain relaxed and keep their eyes open while fixating on a cross. The SCS pulse generators of the patients were configured to a cyclic stimulation program for both the tonic and burst stimulation paradigm. In this cyclic setting, the stimulation alternated between a minute of stimulation and a minute without stimulation, referred to as stimulation ON and stimulation OFF, respectively.

When turning the stimulation on, there was a ramp of one to five seconds before the full pulse amplitude was reached, after which a cycle of stimulation started. The cyclic stimulation protocol settings were determined by balancing data maximization with patient comfort. Due to their chronic pain, patients struggled to remain still in the MEG scanner for extended periods. On the other hand, some patients would fall asleep during longer recordings. One-minute intervals were selected to provide a broad window for capturing acute changes, as the precise onset timing of effects caused by stimulation highly varies between patients [65]. Moreover, these one-minute intervals align with intervals used in a previous fMRI study [32].

### 2.1.3. Recordings

At both institutions, measurements were performed in the same manner. MEG measurements were performed using a 275-channel whole-head CTF MEG scanner in a magnetically shielded room [66]. Data acquisition was done using a sampling rate of 2400 Hz, with a hardware anti-aliasing low-pass filter applied at a cut-off frequency of 600 Hz. The data files were saved with CTF 3rd-order gradient compensation to ensure reliability and integrity of the recorded data. The position of the head in the MEG was registered by marking digital head points on the head in relation to the MEG helmet. In addition to the MEG recording capturing brain activity, other electrodes were used to capture physiological signals, such as cardiac activity and ocular movements. Moreover, an electrode was placed on the back of the patients near the implanted SCS electrode to capture the SCS pulses applied to the dorsal column. For each separate measuring day, a two-minute empty-room recording was performed to capture and assess the instrument and environmental noise in the recording room.

## 2.2. Data Processing

The data processing pipeline for creating the graph-based representations of the resting-state MEG recordings consists of several steps, including data cleaning, data preparation, performing sensor space and source space analysis, epoching of the data, and finally, constructing the graphs that serve as input for the GNN model. An overview of the data processing pipeline is provided in Appendix A.

### 2.2.1. Software

Data preprocessing steps were performed with Brainstorm. Brainstorm is a free, open-source Matlab and Java application for multimodal electrophysiology data analytics and source imaging [67]. Brainstorm is documented and freely available for download online under the GNU general public license. Brainstorm was used with Matlab 2023b. Python was used for further data processing and preparation of input graphs, as well as developing, training, and evaluating the GNN model. A variety of Python's scientific libraries, including MNE-Python, PyTorch Geometric, and Ray were employed [68–70]. All analyses were performed using Python 3.11.11. To integrate the preprocessed data from Brainstorm with Python, an integration step was necessary, which is detailed in Appendix B.

### 2.2.2. Data Cleaning

For data cleaning, a Power Spectral Density (PSD) plot was made for each patient file, using Welch's method with a window of four seconds and 50% overlap. Artifacts could easily be detected in the PSD plots, including power line and stimulation contamination of the signal. Notch filters were used to clean the data of power line contamination by applying filters at the power line's frequency and higher harmonics up to 200 Hz. The power line frequency is country dependent, namely 50 Hz for the Netherlands (the Donders Institute) and 60 Hz for Canada (the MNI). Additional notch filters were used to remove the artifacts at the stimulation frequency and its higher harmonics from all recordings. For this step, the stimulation frequency was assessed for both tonic and burst stimulation in each patient.

Furthermore, the Direct Current (DC) offset is removed from the time series data by applying a Finite Impulse Response (FIR) high-pass filter with a cut-off frequency at 0.5 Hz. This filter removes drifts and offset and passes all frequencies above this threshold.

Additionally, based on the PSD plot for each patient file, bad channels were identified and deleted. Bad channels are characterized by a PSD signal that is significantly deviating from the surrounding sensors throughout the entire frequency band [71]. These channels were marked as bad, removing them from any

further analysis steps. Considerations regarding these preprocessing steps have been detailed in Appendix C.1.

Further data cleaning and filtering steps were omitted, because the removal of artifacts inevitably results in the loss of valuable information alongside noise [72]. Additional data processing steps were implemented in later steps to minimize effects of noise and increase the signal-to-noise ratio. A more comprehensive rationale for omitting further filtering and artifact removal in this step is included in Appendix subsections C.1.3 and C.1.4.

### **2.2.3. Sensor Space and Source Space Analysis**

In this thesis, two distinct approaches for processing MEG data are explored: sensor space analysis and source space analysis. Each approach leads to unique data processing steps, after which the preprocessed data is converted to either sensor-based or source-based graphs.

The sensor space refers to the direct measurements obtained from the MEG sensors, capturing the magnetic fields generated by neural electric currents. In the sensor space analysis, the time series data from each MEG channel is extracted. However, a significant challenge arises due to the variability in sensor placement relative to the patient's head across different recordings. This variability complicates direct comparisons or averaging of sensor-level signals across subjects or sessions, as the correspondence between sensors and specific brain regions is not consistent. Consequently, one sensor does not necessarily correspond to the same brain region within one recording and across multiple recordings [73, 74]. Despite this, the sensor-space approach offers a more straightforward and direct way to construct graph-based representations of MEG data.

To address the sensor space shortcomings, the source space analysis is explored, which provides a more standardized representation of brain activity. The source space is conceptualized as the underlying neural currents that generate the magnetic fields detected by the sensors. This analysis requires a forward model that describes how these neural currents produce the observed magnetic fields, taking into account the anatomical differences between the brain and the MEG sensor array.

#### **2.2.3.1. Sensor Space Analysis**

In sensor space analysis, the filtered time series data collected from the MEG sensors is utilized and exported for further processing steps. This approach involves processing the time series from each channel to identify patterns and compute features directly related to the sensor measurements.

#### **2.2.3.2. Source Space Analysis**

For the source space analysis of brain activity, a model is needed that explains how the neural electric currents, called the source space, produce magnetic fields at external sensors (the sensor space), taking into account the different tissues between the brain and the MEG helmet [75]. This is called forward modelling and results in a head model that describes an approximation of the head geometry based on the patient's anatomy and the locations of the MEG sensors, represented as a set of thousands of vertices. The subsequent process, the source estimation, estimates the brain activity at these vertices based on much fewer sensor locations, called an ill-posed inverse problem [73].

Since individual Magnetic Resonance Imaging (MRI) scans could not be obtained for patients in this study, the template ICBM152 template anatomy was used, and warped (deformed) based on the individual registered head points. For the head model, I used the overlapping spheres model [75]. I performed source estimation using the Minimum Norm (MN) imaging approach, which is recommended due to its computational efficiency and robustness to head model approximations [73]. Consequently, I used the

unconstrained dipole orientation model, where each vertex in the source space was represented by three orthogonal dipoles [73]. A more detailed description of the necessary processing steps in Brainstorm is provided in Appendix C.2.

### ***Regions of Interest***

For the source space estimation, brain regions of interest were defined using a brain atlas, which divides the cortex into anatomically and functionally meaningful areas. In this thesis, groups of vertices representing specific brain regions, called scouts, were defined and created based on the existing Destrieux atlas [76]. Additionally, custom scouts were created or modified to better match the specific regions of interest. The included scouts represented brain regions associated with lateral and medial ascending pathways, and broader cortical areas. These scouts were selected to capture a comprehensive view of whole-brain network dynamics, including specific areas that are associated with pain perception and pain processing [77]. An overview of all scouts is provided in Appendix D.

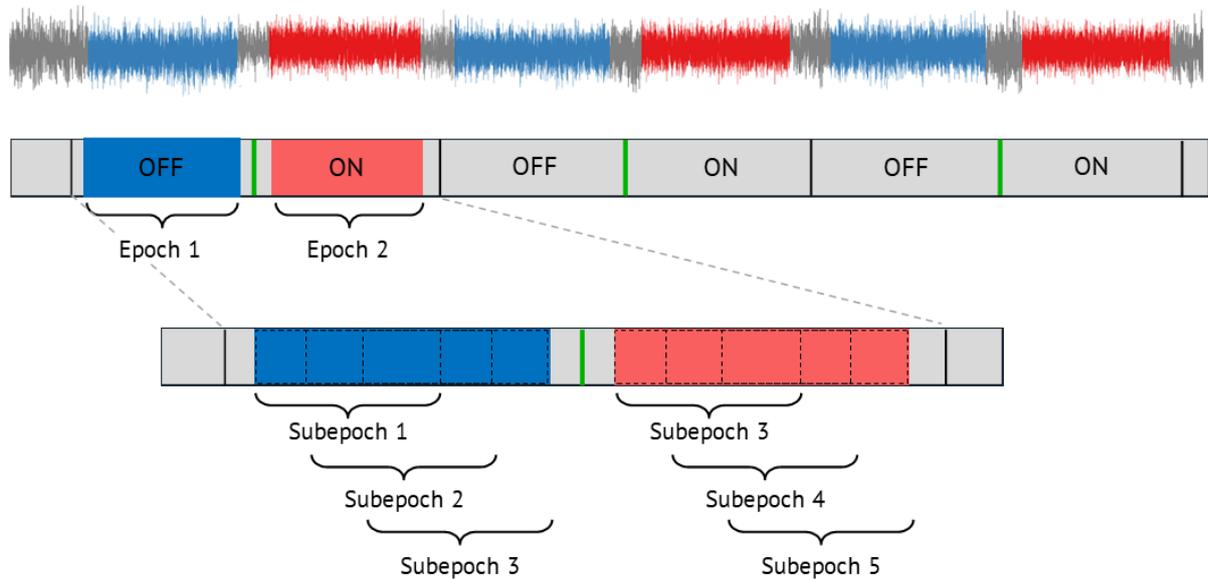
### **2.2.4. Epoching**

Before creating graphs as input for the GNN model, the filtered MEG data is segmented into segments, called epochs, that are labeled based on whether the stimulation was turned on or off. These segments were identified based on the stimulation artifact observed in one of the electrodes prior to filtering. For each recording, I assessed the time points when the stimulation was turned off, thereby retrieving the start and end times of a full cycle. Since there is a delay in reaching the full pulse amplitude when stimulation is turned on, the time points for when the stimulation was turned on were determined using the duration of one stimulation pulse. The recordings were split into 60-second epochs based on the start and end time of each cycle, and labelled with 'stimulation ON' and 'stimulation OFF' consecutively. A ramp time of five seconds was removed at the start and end of each segment, to account for ramping of the stimulation effects, resulting in epochs of 50 seconds. Whenever the stimulation artifact was not visible in one of the electrodes, the patient file was excluded from the dataset, since the timing of the stimulation cycles could not be determined.

Consequently, the segmented data was split into smaller portions to create windows, called subepochs. The subepochs were created using a specified duration (in seconds) and an overlap between consecutive subepochs. An illustration of the epoching process in which the epochs and subepochs are made is shown in Figure 2.1. A careful balance needs to be made for the subepoch length, ensuring that the subepochs capture relevant information without introducing too much redundancy. The optimal configuration of subepoch duration and overlap was found through experimenting with different configurations and evaluating the GNN model performance. This subanalysis has been reported in Appendix E. Based on the results from this subanalysis, the subepoch duration was selected to be 30 seconds with 25 seconds overlap, resulting in 83% overlap between subepochs.

#### **2.2.4.1. Removal of Bad Subepochs**

Some subepochs may still contain noisy data due to motion artifacts, or other sources of interference. To ensure that these segments of data are detected and removed, a method is implemented to detect these artifacts. This is particularly important, since subepochs with artifacts can distort the results and reduce the accuracy of the GNN model. The detection of these artifacts is based on statistical metrics. By calculating the median of the 50-second epoch of the MEG signal and maximum deviation of each subepoch within this epoch, a deviation score was acquired for each subepoch using Equation F.1. This deviation score reflects the extent to which the subepoch deviated from the central tendency of the entire epoch. Subsequently, a threshold was set for this deviation score that determines whether a subepoch should be removed. This threshold was set at 100 for the sensor-based time series, and 300 for the



**Figure 2.1:** The MEG recording with stimulation OFF and ON segments (blue and red segments) is shown at the top. An illustration of this recording shows the same segments and the time points where stimulation was turned off (represented as green vertical lines). Based on these time points, the recording is split into epochs of 50 seconds, excluding a ramping time of 5 seconds at both ends. Consequently, these epochs are split into subepochs with a length of 30 seconds and 25 seconds overlap.

source-based scout data. Appendix F provides a more detailed explanation of this artifact detection approach.

#### 2.2.4.2. Downsampling

Before the graphs are created, the subepoch data is downsampled to reduce the necessary computational power for further analysis steps. The resampling frequency was set to 256 Hz. The Nyquist frequency, defined as half of the sampling rate, represents the highest frequency that can be accurately represented when sampling a signal. After resampling, the Nyquist frequency is 128 Hz. An anti-aliasing filter is applied to prevent aliasing, which can occur when the data is downsampled. Aliasing is the distortion that occurs when high-frequency components of the signal are folded back as lower frequencies due to insufficient sampling rates. This anti-aliasing filter is a low-pass FIR filter with a cut-off frequency at the Nyquist frequency of the resampling rate, in this case 128 Hz, to remove frequencies that may cause aliasing.

## 2.3. Graph Dataset

The GNN model requires a dataset of graphs as input. These graphs are data objects consisting of a representation of the nodes, node features, and edges. For this study, undirected weighted graphs were constructed. Graphs were created for each subepoch of either the sensor space time series data or from the source space scout data.

### 2.3.1. Node Selection

The selection of nodes differs for the sensor-based time series data and source-based scout data. For the sensor-based time series data, the MEG channels are defined as nodes. To ultimately limit the processing time for creating the graphs, only a selection of MEG channels was utilized. This selection of MEG channels was made based on which underlying brain areas would potentially be clinically relevant in pain perception and processing. The selected MEG channels are listed in Appendix G.

Contrarily, for the source-based scout data, the nodes are represented by the different brain sources, as defined in the scout atlas (see Appendix D). As previously mentioned, the implemented scouts have been selected based on existing brain atlases and previous research.

Both sensor- and source-based node definitions have been utilized in previous research on GNN models for brain connectivity-based graph classification tasks [60, 78, 79]. Since the number of nodes directly influences graph size, and larger graphs demand greater computational resources, a balance was struck to ensure the inclusion of relevant brain regions while avoiding excessively large input graphs. For the sensor-based graphs, 46 MEG channels were selected as nodes. For the source-based graphs, 33 scouts were used, resulting in 33 nodes.

### 2.3.2. Node Feature Computation

In this study, the node feature matrix was derived from the PSD, which quantifies neural oscillatory activity across different frequency bands. For each subepoch, the PSD values at each node were computed across multiple frequency bins, representing the power of neural activity within a specific frequency range. This node feature definition has been previously utilized in various studies that developed GNN models for brain connectivity analysis [56, 58, 78, 80]. The choice of this node feature was motivated by its ability to capture regional neural activity by highlighting the power of distinct oscillatory rhythms, including the delta, theta, alpha, beta, and gamma bands. These frequency band-based graph features can provide insights into different aspects of brain functioning, since each band is associated with specific physiological and cognitive processes [81]. Moreover, compared to using the time series signals as node features, the PSD offers a computationally efficient alternative that substantially reduces dimensionality while retaining key spectral characteristics. By leveraging PSD features, the GNN model could focus on meaningful patterns in neural activity without the added complexity of high-dimensional input data.

The PSD was computed as node feature for each subepoch of MEG data using Welch's method. A window size of one second with 50% overlap was applied, resulting in a frequency resolution of 1 Hz. The length of the subepoch determines the number of windows included in the PSD computation. Since Welch's method averages the PSD across all windows, longer subepochs provide a more stable and reliable PSD estimate due to increased averaging. Additionally, larger subepochs help mitigate the effects of transient noise and artifacts more effectively than shorter subepochs.

Before computing the PSD for each subepoch and node, a frequency band of interest was defined. For the initial development, training, and optimization of the GNN model, the 1 - 100 Hz range was used, referred to as the full band.

Beyond this full frequency range, the model was also trained and evaluated on five distinct frequency bands to assess performance variations across different oscillatory components of the MEG signal. The specific frequency bands analysed are the normative frequency bands described by Groppe et al. [82], and are presented in Table 2.1. Additionally, a frequency band containing both the theta and alpha frequency ranges (4 - 12 Hz) is included, based on previous research that found significant changes in theta and alpha activities during the activation of electrical nerve stimulation and SCS [83, 84]. Merging these two frequency bands offers the advantage of increasing the number of frequency bins in the node feature matrix, potentially enhancing the model's ability to capture relevant spectral patterns within this frequency band.

Since the PSD values differ in magnitude across different patients, a normalization step was necessary to make the node features more comparable across different subepochs and files. This step ensures that the model can focus on relative changes in the PSD, rather than the absolute values, which can vary due to

**Table 2.1:** Frequency bands used for model training and evaluation.

Frequency Band (Hz)	Band Name
1 – 100	Full Band
1 – 4	Delta
4 – 8	Theta
8 – 12	Alpha
4 – 12	Theta & Alpha
12 – 30	Beta
30 – 100	Gamma

different recording conditions or noise. This normalization was based on baseline correction, where the average PSD values from stimulation OFF epochs for each patient was selected as baseline, as described in Equation H.1. This node feature normalization is further elaborated in Appendix H. Consequently, for each frequency band, a node feature matrix is constructed using the normalized PSD values for every node and subepoch of MEG data, resulting in one node feature matrix per graph. Ultimately, separate graph datasets are created for all frequency bands, allowing a GNN model with the same architecture to be trained on each graph dataset independently.

### 2.3.3. Edge Definition and Edge Weight Computation

The edges of the input graphs were defined using a connectivity matrix, which was composed by computing a connectivity metric between each pair of nodes from the sensor-based or source-based signals. An edge between a pair of nodes exists if the resulting connectivity metric is a non-zero value. The number of edges for an undirected graph can be retrieved using Equation 2.1, where  $E$  is the number of edges, and  $N$  is the number of nodes. Using the sensor-based data, 46 nodes will result in 1035 edges. The 33 nodes from the source-based data will result in 528 edges.

$$E = \frac{N * (N - 1)}{2} \quad (2.1)$$

The edge has a weight attribute, assigning a weight to the connection between a pair of nodes, that is defined as the absolute value of the connectivity metric. Previous research on the application of GNN models for brain connectivity classification tasks has demonstrated that functional connectivity metrics tend to outperform anatomical connectivity metrics, such as the Euclidean distance between spatial positions of EEG channels [58]. This discrepancy may be attributed to the fact that functional connectivity captures both short- and long-range interdependencies between neural signals, whereas spatial distance metrics are limited to local, physical relationships and fail to reflect long-range functional interactions across EEG or MEG channels [85]. As a result, functional connectivity-based graphs may provide a more comprehensive representation of both local and global brain network dynamics. Additionally, while effective connectivity offers insights into directional influences between brain regions, it inherently requires the use of a directed adjacency matrix. This poses a challenge for standard GNN architectures, which typically assume undirected graphs. To address this, a directed graph convolution operation has been proposed by Tong et al. [86], enabling the modelling of directed relationships. However, this method introduces increased computational complexity and may require more extensive training resources.

For this study, I selected a functional connectivity metric for computation of the edge weights, as it offers a more comprehensive representation of brain network dynamics while avoiding the challenges associated

with directed graph convolutions. The connectivity metric that was used in this thesis was the Phase Lag Index (PLI). This measure was introduced by Stam et al. and is computed using a function from MNE-Python [87, 88]. A more detailed explanation on the PLI as functional connectivity metric and the calculations is provided in Appendix I.

The PLI was selected due to its ability to capture true neuronal interactions while minimizing the confounding effects of volume conduction and common sources. Volume conduction is the passive spread of electrical or magnetic signals through conductive tissues, creating false correlations between sensors. Since PLI focuses on consistent non-zero phase differences between signals, the zero-phase lag connections, that are often spurious and attributable to volume conduction, are discounted. This property ensures that the connectivity measures more accurately reflect true functional interactions between brain regions. For a phase difference fluctuating around zero, a low PLI value will be calculated. Hence, a PLI value of zero indicates either no coupling or coupling with a phase difference centred around zero, which is typically considered to reflect non-true (e.g., volume conducted) connectivity. A high PLI value, on the other hand, reflects consistent non-zero phase lags between two time signals, suggesting genuine functional connectivity that is less likely to be explained by volume conduction or common sources.

Furthermore, PLI is applicable across various frequency bands, which is advantageous given that different oscillatory networks are associated with distinct aspects of pain processing and neuromodulation [84, 89, 90]. Similarly to the PSD computation, the frequency bands in Table 2.1 were applied for PLI computation. As mentioned previously, separate datasets of input graphs were constructed for each frequency band, resulting in multiple GNN models with the same model architecture that were trained on these different graph datasets.

Despite its advantages, the PLI also has certain limitations as a connectivity metric. The PLI evaluates the asymmetry of the phase difference distribution and, as a result, detailed information about the magnitude of phase differences is lost, potentially reducing the sensitivity to subtle but meaningful variations [91]. Additionally, the PLI exclusively measures phase synchronization and does not account for amplitude-based coupling, limiting its ability to capture interactions that involve both phase and amplitude dynamics.

Nevertheless, connectivity matrices were constructed using the PLI for each subepoch of MEG data, representing functional connectivity between all nodes within that subepoch. While these matrices for each subepoch served as input to the GNN model, an additional matrix representing the difference between stimulation ON and OFF states was also computed to explore connectivity changes potentially relevant to SCS effects.

## 2.4. Graph Neural Network Model

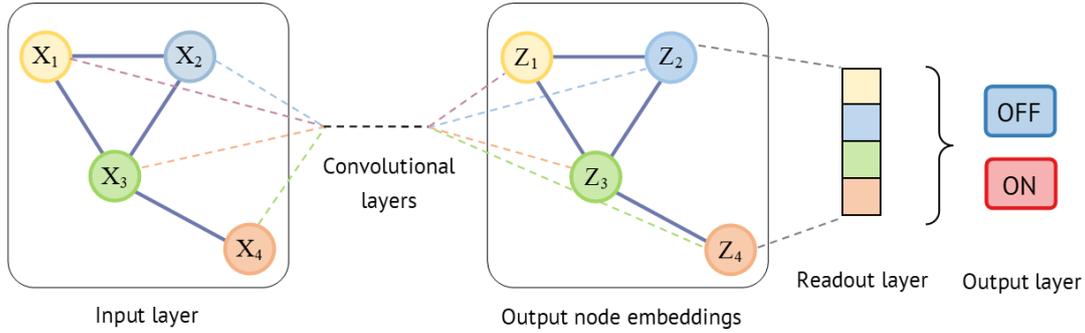
The GNN model used in this thesis is designed to classify MEG data into stimulation ON and OFF states. Since the input of the GNN model is the dataset consisting of graphs with labels of either stimulation ON or OFF, this model is created for graph classification.

### 2.4.1. Model Architecture

The model architecture consists of an input layer, multiple graph convolutional layers, a readout layer, and a final linear layer for classification. Figure 2.2 provides a visual representation of the GNN model architecture. The architecture is defined as follows:

1. **Input Layer:** The input to the model is a graph with node features, edges, and edge weights.

2. **Graph Convolutional Layers:** The model includes multiple graph convolutional layers, in which the node features are iteratively updated to acquire updated node embeddings.
3. **Readout Layer:** A global mean pooling layer is used to aggregate node embeddings into a graph-level representation.
4. **Output Layer:** The final linear layer maps the graph-level representation to probabilities for each of the output classes (stimulation ON or OFF).



**Figure 2.2:** Visualization of the GNN model, including the input graph with initial node features ( $X_n$ ) for each node  $n$ , the convolutional layers, the output graph with updated node embeddings ( $Z_n$ ), a readout layer, and the output classification, consisting of class probabilities.

#### 2.4.1.1. Input Layer

The input for the GNN model consists of the dataset of created graphs with labels for either stimulation ON or OFF. From the graphs, the node feature matrix, the edge indices, and the edge weights are retrieved to serve as input for the model.

#### 2.4.1.2. Graph Convolutional Layer

A core part of the GNN model is the graph convolution layer, in which the model learns the graph characteristics through iterative updating of the node feature representations. The convolutional layer that is implemented is the graph convolutional operator proposed by Kipf and Welling [92]. This convolutional operator belongs to the spectral GNNs, which are designed to perform spectral domain analysis of graph-structured data. The spectral convolution operation is defined in the Fourier domain by computing the eigendecomposition of the graph Laplacian matrix [93]. The approach proposed by Kipf and Welling provided a simplified approach based on the spectral convolution operations and proposed the layer-wise propagation rule as shown in Equation 2.2, where the node feature matrix is represented as  $X$ , and the updated node feature representation matrix is  $Z$ .

$$Z = \sigma(\tilde{A}XW) \quad (2.2)$$

In this equation, the normalized adjacency matrix  $\tilde{A}$  is a degree-adjusted representation of the graph's structure. For an undirected graph, the degree matrix  $D$  is a diagonal matrix, where each diagonal element represents the sum of the number of connections for each node. Through adjusting the adjacency matrix ( $A$ ) based on the degree of each node, the normalized adjacency matrix better reflects the connection strengths between each node, where the amount of connections to other nodes is not influencing the connection strength [94]. Since the process of updating the node features takes both the information from neighbouring nodes, as well as the node's own information, a self-loop is implemented in the adjacency matrix using the identity matrix  $I_N$ . The normalized self-looped version of the adjacency matrix is defined using Equation 2.3.

$$\tilde{A} = I_N D^{-1/2} A D^{-1/2} \quad (2.3)$$

The trainable weight matrix  $W$  in Equation 2.2 is randomly initialized before model training and is updated during the training process to refine the node representations, ultimately enabling improved classification performance. Lastly,  $\sigma$  is the non-linear Rectified Linear Unit (ReLU) activation function.

The ReLU function is a commonly used activation function in deep learning models. The activation function in a neural network is responsible for ensuring that consequent layers learn larger-scale network interactions inside the graphs by introducing nonlinearity at each propagation. The ReLU function is defined as  $ReLU(x) = \max(0, x)$ , meaning that negative input values are set to zero, ultimately reducing the impact of less useful feature contributions. Without the ReLU function, each layer is a linear matrix multiplication  $Z = \tilde{A}XW'$ , where  $W' = W^{(1)}W^{(2)}W^{(3)}\dots W^{(L)}$  are the trainable weight matrices for each layer, with  $L$  being the total number of convolutional layers. This is not desired since each layer's contribution does not introduce additional expressive power in this matrix multiplication. Hence, the ReLU function is implemented to force each layer to learn new patterns. Since deeper layers may capture large-scale network interactions, the ReLU function allows the model to learn more complex relationships within the graph structure.

After each convolutional layer, batch normalization is implemented in the GNN model to stabilize training and speed up convergence by normalizing layer inputs. During model training, the model is iteratively updating its parameters based on a set of graphs that are inputted, called a batch. Batch normalization computes the mean and variance of all node features across all graphs in the batch, and uses this to normalize the features. This ensures that the updated node feature distributions stay stable across layers.

Consequently, a dropout layer is implemented with the aim to prevent overfitting. The dropout layer randomly selects a fraction of the updated node features  $Z$  and sets them to zero. This is controlled by a dropout probability, which describes the fraction of node features that are dropped in each training iteration. Dropout helps by introducing randomness, preventing the model from relying too much on specific features or connections.

#### 2.4.1.3. Readout Layer

The readout layer in a GNN transforms the node-level feature representations into a graph-level representation. In the GNN model, global mean pooling is utilized as readout layer. Global mean pooling computes the mean of the updated node features for all nodes in a graph, as described in Equation 2.4, where  $x_i$  represents the node features of node  $i$ , resulting in a graph-level embedding  $x_G$ .

$$x_G = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.4)$$

#### 2.4.1.4. Output Layer

The final layer of the GNN model is a linear layer, which acts as an affine transformation to map the learned graph-level representation to the desired output space. This layer outputs a vector of class scores (logits) for each graph, representing the model's confidence in each class. The classification decision is then determined by selecting the class with the highest logit value, ensuring that the model assigns the most probable label based on the learned graph representations.

## 2.4.2. Model Training

### 2.4.2.1. Data Split

After the creation of the graph dataset, the graphs are split into a training and test set. A stratified split is applied to partition the graphs, ensuring that the distribution of labels is maintained in both the training and testing sets. This split creates a training set of 80% and a test set of 20% of all patient files that are used as input.

For the train set, the total number of graphs of both labels is retrieved, called the class counts. Subsequently, class weights were calculated and included in the model training process to account for any potential class imbalance. In this study, the risk of class imbalance is increased since the class is dependent on the epochs (50-second windows) that are segmented in the MEG recording. Since the MEG recording might not fully capture the epochs at the start and end of the recording, unequal amounts of epochs with stimulation ON and stimulation OFF may be present. This will ultimately result in a class imbalance.

Before training the model, the training set is split into training and validation subsets, allocating 80% for training and 20% for validation. Similarly to the train-test split, a stratified split was used to maintain the original class distribution in both subsets. The validation set is utilized to evaluate the training process.

In this study, I initially focused on training and optimizing model settings using sensor-based data within the 1 – 30 Hz frequency range, as certain frequencies above 30 Hz were removed via notch filtering, potentially affecting the model to become less generalizable across patients, as different segments of the frequency spectrum are removed for each individual. After setting up this initial model, I trained and further optimized the model using sensor-based data from the full frequency band (1 - 100 Hz).

### 2.4.2.2. Training Procedure

The model is trained over 100 training iterations. In each training iteration, batches of the training and validation set are created. Through batching of the data, the graphs inside the set are essentially grouped together with a certain size, referred to as the batch size, to facilitate efficient data loading.

To train the GNN model for the graph classification task, the cross-entropy loss function was used to quantify the discrepancy between the outputted class probabilities and the true class labels. For each graph input, the model outputs a probability distribution over the possible classes, and the cross-entropy loss penalizes incorrect predictions by assigning a higher loss when the predicted probability for the true class is low. The cross-entropy loss is computed using Equation 2.5, where  $p_i$  represents the probability for class  $i$ ,  $y_i$  represents the true class (one for the correct class, zero for the incorrect class), and  $\omega_i$  represents the class-specific weights. The class weights are implemented as a weighing factor to mitigate the effects of class imbalance.

$$L = - \sum_i \omega_i * y_i \log(p_i) \quad (2.5)$$

The loss was minimized using backpropagation in combination with the Adam optimizer, an adaptive learning rate optimization algorithm [95]. Gradients represent the rate of change of the loss with respect to each model parameter, indicating how much a small change in a parameter will affect the model's error. Backpropagation computes these gradients by propagating the error backward through the network. The Adam optimizer adjusts each model parameter based on the average direction and size of recent gradients, helping the model learn more efficiently. By considering both the magnitude of past gradients (how much the loss changes) and the variability of those changes, the Adam optimizer assists in adapting the model parameters, thereby improving the model's convergence and performance.

Thus, during each training step, the gradients of the cross-entropy loss with respect to the model parameters were computed, and the Adam optimizer used these gradients to update the parameters in a direction that minimizes the loss. The updated parameters were then evaluated using the training and validation set after each training iteration. The set was inputted to the model in batches of the same batch size. Over all batches, the training and validation accuracy were computed. Since the training set was directly used to update the model's parameters, the training accuracy may be optimistically biased and not fully reflective of the model's generalization ability. Therefore, the validation accuracy serves as a more reliable metric for monitoring the model's performance over training iterations, providing insight into how well the model generalizes to unseen data. This iterative optimization process enabled the model to gradually learn discriminative features. After one training iteration, the gradients are cleared, and the training and validation loss are set to zero. After all training iterations are completed, the model's performance on both the training and validation set is reported.

#### **2.4.2.3. Early Stopping**

Overfitting is a large risk in deep learning models, where the model becomes too specific on the training set by learning noise or specific details that do not generalize well to unseen data. Hence, monitoring the training and validation loss over the training iterations, referred to as loss curves, is essential. A decreasing training loss curve iterations is indicative of the model that is learning and updating its parameters accordingly. The validation loss over the training iterations can provide insights into possible overfitting of the model. A decreasing validation loss curve indicates that the model is generalizing well and improving its performance on unseen data. However, if the validation loss curve decreases, but then increases again, while the training loss curve is still decreasing, this is a sign of an overfitting model. The model is learning too well on the training set, and failing to generalize to unseen data.

To prevent overfitting, early stopping was implemented in the GNN model [96]. The validation loss was monitored throughout the training process, and the loss did not decrease for 10 consecutive training iterations, training was halted. Before the early stopping could be initiated, a minimum of 30 training iterations had to be passed.

#### **2.4.3. Hyperparameter Optimization**

Hyperparameter optimization is a critical step in training machine learning models, as it involves tuning the model's hyperparameters to achieve optimal performance on the classification task. In this thesis, various hyperparameter configurations were explored for the GNN model. To ultimately identify the best combination of hyperparameters that maximizes the model's accuracy on the validation set, hyperparameter tuning was performed using a grid search for the different parameters. A number of configurations for all parameters was implemented, and for each configuration of parameters, a trial was created. For each trial, the data split into training and validation sets was used to train and evaluate the model, with a unique configuration of parameters. After all configurations finished training, the best-performing hyperparameter configuration was selected based on the highest validation accuracy.

The hyperparameters that were explored are visualized in Table 2.2. The parameters involve parameters for edge filtering, as well as model-specific parameters. Hyperparameter values were chosen based on commonly described values in literature, clinically relevant values, and empirical evidence regarding computational barriers.

##### **2.4.3.1. Edge Filtering Parameters**

Edge filtering is a technique that can be used to refine the graph structure by removing or selecting specific edges, with the aim of simplifying the graph or focusing on important connections through removing

**Table 2.2:** Different hyperparameter settings that were evaluated.

Hyperparameter	Settings
<i>Edge Filtering Parameters</i>	
Threshold	None, 0.01, 0.03, 0.05, 0.07
Top K	None, 300, 600, 900
<i>Model-specific Parameters</i>	
Layers	2, 3, 4
Hidden Channels	16, 32, 64, 128
Batch Size	2, 4, 8, 16, 32, 64, 128
Learning Rate	0.00001, 0.0001, 0.001, 0.01
Dropout Rate	0.01, 0.1, 0.3, 0.5

spurious connections in the graph. Edge filtering is applied before the graphs are inputted to the GNN model. This process encompasses two different approaches to edge filtering, threshold filtering and top-K filtering, which have been utilized in previous research [58, 97]. For these approaches, the assumption is made that the edge weight, i.e. the connectivity strength, is directly related to the importance of the edge.

#### ***Threshold Filtering***

Threshold filtering is the first step in refining the graph structure. In this process, a predefined threshold value is used to remove weak connections between nodes. Specifically, only edges with a weight greater than the threshold are retained, while all others are discarded.

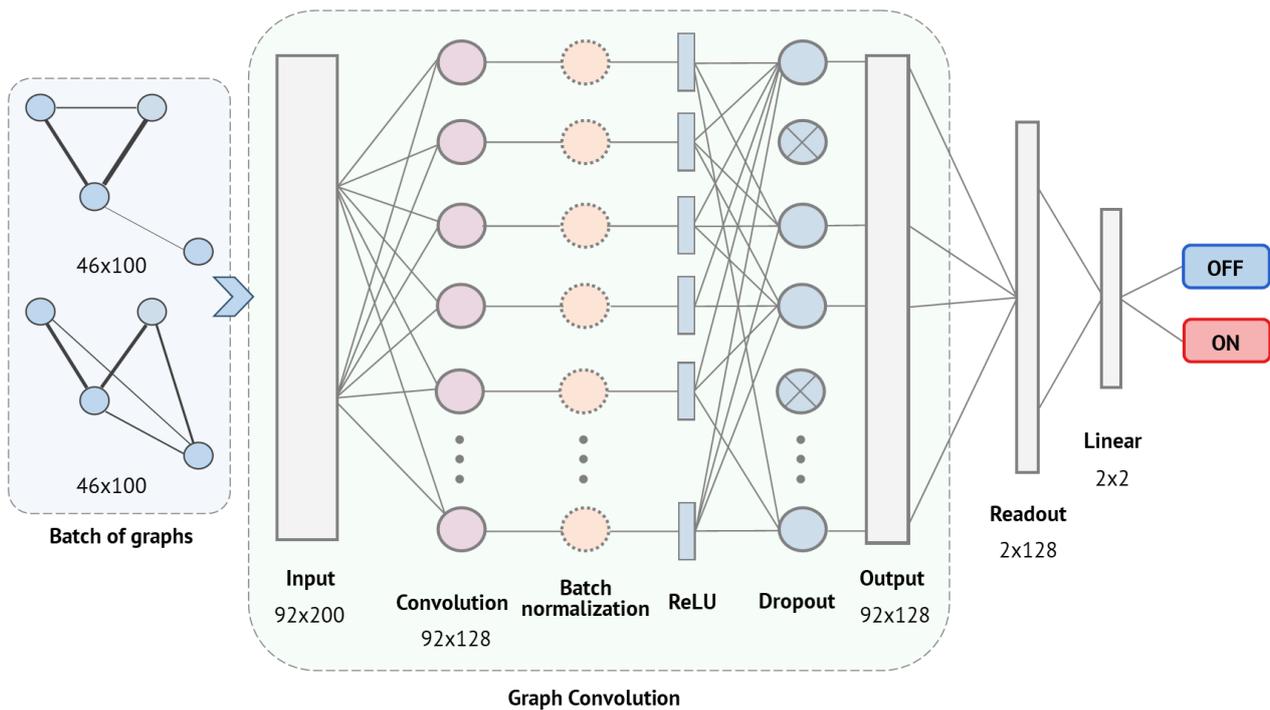
#### ***Top-K Filtering***

After this, top-K filtering is applied to further refine the graph structure. In this step, only the top K edges with the highest weights are retained for the entire graph. This involves sorting the edges by weight and selecting the K strongest connections. Top-K filtering ensures that the graph remains sparse and focused, highlighting the most significant relationships while eliminating weaker and potentially redundant edges.

#### **2.4.3.2. Model-specific Parameters**

Six model-specific parameters were optimized to determine the best configuration. These parameters were chosen for their impact on the model's ability to extract relevant features, avoid overfitting, and effectively learn from the input graphs.

Figure 2.3 provides an overview of the layers within the GNN model architecture with one layer, along with several model-specific hyperparameters. The model input consists of a group of graphs, called a batch, in which each graph is represented by a node feature matrix and an adjacency matrix. In the graph convolutional layer, the entire batch is processed to update the node features into updated node embeddings. This transformation maps the original features to a specified number of hidden channels, which defines the dimensionality of the embeddings. This is followed by batch normalization, a ReLU activation function to introduce non-linearity, and a dropout layer, which randomly deactivates a fraction of the node features during training based on a set dropout rate. Multiple convolutional layers can be stacked, depending on the depth of the GNN model, repeating the same graph convolution process with as input the updated embeddings from the previous layer. After all graph convolutional layers, a readout operation aggregates the node embeddings into a single embedding per graph. A final linear layer then maps these graph-level embeddings to the two output classes, producing a prediction based on the model's confidence for each class.



**Figure 2.3:** Overview of a GNN model architecture for one layer with several model parameters. Input graphs with 46 nodes (visualized as graphs with four nodes for clarity) and 100 node features (corresponding to 100 frequency bins) are batched with a batch size of two, resulting in a total input shape of  $92 \times 200$ . After convolution, the updated node embeddings are transformed to the number of hidden channels. For 128 hidden channels, the node embeddings have a shape of  $92 \times 128$ . After the convolution mechanism, batch normalization is applied, after which a ReLU activation function introduces non-linearity, and subsequently, a dropout layer is applied. The output of the graph convolutional layer has a shape of  $92 \times 128$ . The readout layer transforms the updated node embeddings into a shape of  $2 \times 128$ , forming one embedding for each graph. The linear layer maps these embeddings to the number of output classes, resulting in a shape of  $2 \times 2$  for this batch of two graphs and two output classes (ON and OFF).

### Layers

The number of layers in the GNN determines the depth of the model by specifying the number of convolutional layers, indicated as the green coloured block in Figure 2.3. Each layer represents a single step of node feature aggregation and transformation, as described in Equation 2.2. Therefore, the number of layers is essential for the depth at which information is propagated across the graph. Increasing the number of layers allows the model to capture more complex relationships by aggregating information from farther nodes. However, deeper networks can suffer from over-smoothing, where node representations become nearly identical. This occurs because, with each additional layer, information is aggregated from an increasingly larger neighbourhood. Especially in relatively smaller graphs, this can result in features from all nodes influencing each other, eventually making the node embeddings indistinguishable. Ultimately, this introduces a larger risk of overfitting, particularly with limited training data.

### Hidden Channels

The number of hidden channels is a parameter that defines the number of feature dimensions of the updated node feature matrix, that is the output of the convolution operation. The number of hidden channels determines the capacity of the model to learn distinctive features. More hidden channels allow the model to learn more expressive node embeddings, capturing complex patterns in the data. However, increasing the number of hidden channels also increases computational cost and the risk of overfitting. A range of values for the hidden channels was implemented and evaluated to determine an optimal

trade-off between model expressiveness and efficiency.

### ***Batch Size***

The batch size defines the number of graphs processed together during model training. A larger batch size can lead to more stable gradient updates and faster convergence but requires higher memory usage. Smaller batch sizes introduce more noise into the gradient updates, which can sometimes help generalization but may lead to instability during training.

### ***Learning Rate***

The learning rate controls the rate at which the model updates its parameters in response to the computed loss for each training iteration. A smaller learning rate provides more precise convergence but can slow down training. Conversely, a larger learning rate speeds up training but may lead to divergence if updates are too large. A range of learning rates was explored to optimize convergence stability while maintaining efficient training.

### ***Dropout Rate***

Dropout is a regularization technique that randomly deactivates a fraction of node features during model training to prevent overfitting and to improve generalization of the model [98]. Dropout layers work by randomly selecting a set of node features that are 'dropped' by setting these features to zero, which forces the network to learn on the remaining graph. A dropout rate is assigned that describes the fraction of node features that is randomly set to zero during training. This ensures that the GNN model does not rely too heavily on specific node features, thereby preventing overfitting and improving generalization. A higher dropout rate increases regularization but may lead to underfitting, while a lower rate may not provide sufficient regularization.

#### **2.4.3.3. Hyperparameter Scheduler**

The AsyncHyperBandScheduler was employed to efficiently allocate computational resources to the most promising hyperparameter configurations [99]. This scheduler implements the asynchronous successive halving technique to accelerate the tuning process, through asynchronously eliminating underperforming trials early and allocating resources to promising configurations. Moreover, the asynchronous nature of the scheduler allows it to handle multiple trials in parallel, making it suitable for large-scale hyperparameter optimization tasks [100].

As trials are executed, underperforming trials are identified by monitoring the validation accuracy at each training iteration. After each training iteration, the scheduler prunes the bottom-performing trials, only retaining the best-performing trials, allowing more computational resources to be allocated to the better-performing trials. The number of trials that are discarded during model training is variable, as this is dependent on observed performance, and poorly performing trials are stopped early once sufficient evidence is collected [101].

#### **2.4.4. Model Evaluation**

To evaluate the model outcomes, I implemented various strategies, including monitoring the training process with accuracy and loss curves, as well as reporting classification metrics after the model had finished training.

During model training, both training and validation accuracy and loss were tracked to assess the convergence process of the model. Curves for these parameters were plotted to monitor the training process and identify any potential overfitting. The training loss curve represents how well the model is learning from the data. Ideally, the training loss should decrease over each training step, indicating that the model is

learning and outputting better results. The same loss curve was plotted for the validation set, which is a separate dataset that was not used for model training. A decreasing validation loss curve indicates that the model is generalizing well and improving its performance on unseen data. However, if the validation loss curve decreases, but then increases again, while the training loss still decreases, this is indicative of an overfitting model. An overfitting model is learning too well on the training dataset and failing to generalize to unseen data. Hence, these loss curves provide insights into the model performance, the occurrence of overfitting or underfitting, and can be used as a guide to adjust model parameters.

Model performance was evaluated using different input configurations for the GNN, with accuracy, F1-score, Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) serving as the primary evaluation metrics. Accuracy offers a straightforward and intuitive measure of the model's ability to distinguish between classes and is calculated using Equation 2.6, where True Positives (TP) and False Positives (FP) correspond to graphs classified as stimulation ON, and True Negatives (TN) and False Negatives (FN) correspond to graphs classified as stimulation OFF. To complement this, the F1-score was computed using Equation 2.7, serving as a harmonic mean of precision (positive predictive value) and recall (sensitivity). This metric is especially valuable when dealing with potential class imbalance, which may be relevant in the current study [102].

The ROC analysis is a commonly used tool to evaluate binary classification models by providing insight into how well a model distinguishes between two classes. The ROC curve is generated by plotting the True Positive Rate (sensitivity) against the False Positive Rate (1 - specificity) across a range of classification thresholds. This provides a comprehensive view of the model's behaviour under different decision boundaries. The AUC measure summarizes this ROC curve into a single value ranging from 0 to 1, where an AUC of 1.0 indicates perfect separation between the classes. An AUC of 0.5 suggests the model performs no better than random guessing and values below 0.5 imply a systematic misclassification [103]. One of the main advantages of ROC AUC is that it is threshold-independent, since it evaluates the model's performance across all possible thresholds rather than relying on one arbitrary cut-off. This makes it particularly useful in early model development and model comparison. Additionally, ROC AUC is robust to class imbalance.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.6)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (2.7)$$

For a comprehensive evaluation, both validation and test accuracy were reported, with validation accuracy monitored during the training process. The F1-score and AUC based on the test set were also reported to provide a more robust measure under imbalanced class conditions. Furthermore, ROC curves for the different GNN model configurations were assessed to compare the models.

Additionally, confusion matrices for the test set were analysed to gain further insight into classification behaviour, offering an overview of TP, FP, TN, and FN values.

#### 2.4.4.1. Sensor-based GNN model

Model evaluation was conducted by varying the inputs for the GNN model and assessing the model performance for each scenario. Initially, sensor-based data across different frequency bands were evaluated, including data from the full frequency band (1 - 100 Hz), as well as the distinct frequency bands (delta, theta, alpha, and gamma bands). Through analysing individual frequency bands, the spectral features

that contribute most to classification performance can be identified, providing insight into the neural oscillations underlying pain processing and SCS effects. Since different frequency bands are associated with distinct cognitive and sensory functions [104], this approach helps determine whether specific bands carry more relevant information than others.

Additionally, sensor-based data from different stimulation paradigms were considered. The tonic and burst stimulation paradigms were each evaluated separately using MEG recordings from the full frequency band. Tonic and burst stimulation differ in their temporal patterns of electrical pulses, potentially leading to different effects on brain connectivity and pain modulation [12–14]. By analysing both recordings separately, it can be assessed whether the GNN model distinguishes specific neural responses unique to each paradigm. This insight may contribute to a better understanding of the underlying mechanisms of the SCS paradigms.

Further evaluation was conducted by considering sensor-based data from both recording locations independently to assess whether any variation in the data is attributable to site-specific conditions. This approach helps to ensure that the model's performance is not confounded by location-related factors, providing a more robust evaluation of the underlying neural dynamics. This included MEG recordings from the Donders Institute and the MNI, with data from the full frequency band.

To evaluate the model's generalizability, an external test set was used on a previously trained model. Specifically, two types of generalization were assessed: cross-paradigm and cross-site performance. For cross-paradigm evaluation, the sensor-based GNN model trained on tonic stimulation data was tested on burst stimulation data, and vice versa, to examine how well the model transfers across stimulation types. For cross-site evaluation, the model trained on data from the MNI was tested on recordings from the Donders Institute, and vice versa, to assess performance on an unseen patient set.

#### **2.4.4.2. Source-based GNN model**

Lastly, source-based data from the full frequency band was also used as input to the GNN, and the model performance was compared to that of the model trained on sensor-based data from the full frequency band. Evaluating both approaches allows for determining whether source-based graphs enhance classification performance and improve the interpretability of the model's learned representations.

### **2.4.5. Node Explainability**

The goal of this study was to gain insight into the active brain networks in chronic pain patients with SCS through the GNN model. However, GNNs are inherently complex models with a "black-box" nature, making it difficult to interpret how the decisions are made. Understanding which specific brain regions are most influenced by SCS is crucial for both clinical application and model optimization.

#### **2.4.5.1. SubgraphX**

To address this challenge, explainability techniques were explored to increase the interpretability of the GNN model's decisions [105]. Specifically, SubgraphX, an explainability framework proposed by Yuan et al. [62] designed for node and graph-level interpretation, was implemented to identify the most influential nodes in the graph. By systematically exploring various combinations of nodes and their corresponding node features, the explainability method identifies the substructures that most significantly influence the model's predictions. This process helps highlight the key brain regions that are critical in distinguishing between different patient states. The methodology of SubgraphX and the implementation in this study is detailed in Appendix J. For each inputted graph, the explainability method provides an explanation consisting of a maximum of five nodes forming the identified subgraph and two key metrics: the fidelity and sparsity scores.

### 2.4.5.2. Fidelity and Sparsity Scores

The fidelity score measures the impact of removing the identified subgraph on the model's performance, quantifying how much the model's predictions change without this subgraph. The sparsity score indicates the proportion of nodes used to explain the model's prediction relative to the total number of nodes, providing insight into how much of the graph is necessary for the explanation.

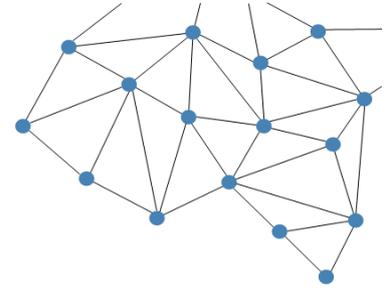
### 2.4.5.3. Node Explainability Analysis

Through the implementation of SubgraphX, it was possible to identify which nodes (MEG sensors or cortical sources) significantly influenced the GNN's decision-making process. This method was applied on both sensor-based and source-based trained GNN models (using the full frequency band), which were tested on MEG recordings with both SCS paradigms (tonic and burst) from one patient. Patient *PT06* was selected for this analysis due to the relatively longer recording durations compared to other patients, which provided a larger set of graphs for the explainability analysis and contributed to more robust results. Additionally, this patient reported noticeable differences in sensation between the stimulation ON and OFF states, suggesting that the SCS had immediate effects on pain relief, with pain re-emerging when the stimulation was turned off.

The sensor-based and source-based GNN models have different input nodes, and therefore, the outcomes of the node explainability analysis will show each model's learned representations and decision-making focus, reflecting the differentiating characteristics of its respective input data. As these models were tested and analysed using recordings from both SCS paradigms, the explainability analysis can pinpoint the most important nodes for each stimulation type. This aims to determine whether different nodes play a key role in classifying stimulation states in tonic versus burst stimulation.

### 2.4.5.4. Node Explainability Outcomes

Since the explainability method generates an explanation for each graph in one patient file, a comprehensive explanation was obtained by compiling a list of the five most frequently highlighted nodes across all subgraphs. The counts of how often each of the five nodes were identified were reported relative to the total number of graphs, reflecting their consistency in contributing to the model's decisions. Additionally, the average fidelity score over all identified subgraphs was calculated, along with the sparsity score for each individual explanation. This approach allows for a broader understanding of which brain regions consistently contribute to the model's predictions of this file and how the model's decision-making process varies across different graphs.



# 3 | Results

## 3.1. Patient Population

Twenty-two patients treated with SCS were enrolled in this study. The patient characteristics (Table K.1) indicate the variety of pain duration, the different locations of pain, as well as the range in NRS pain scores with both stimulation paradigms. The majority of the patients (18/22) suffered from Persistent Spinal Pain Syndrome type 2 (PSPS-2), which is chronic pain that persists after spine surgery. Other pain conditions were diabetic neuropathy and Complex Regional Pain Syndrome (CRPS). All patients suffered from back pain or pain in the lower extremities. The effectiveness of the SCS was different for each patient.

One patient only completed the MEG recording session with the burst stimulation paradigm. Eight MEG recordings, originating from six unique patients, were excluded due to the absence of detectable stimulation artifacts, which prevented the identification of stimulation cycles. A total of 20 patients and 35 individual MEG recordings were included for the analysis. An overview of all recordings, including the excluded recordings, and the assessed stimulation frequencies is shown in Table K.2.

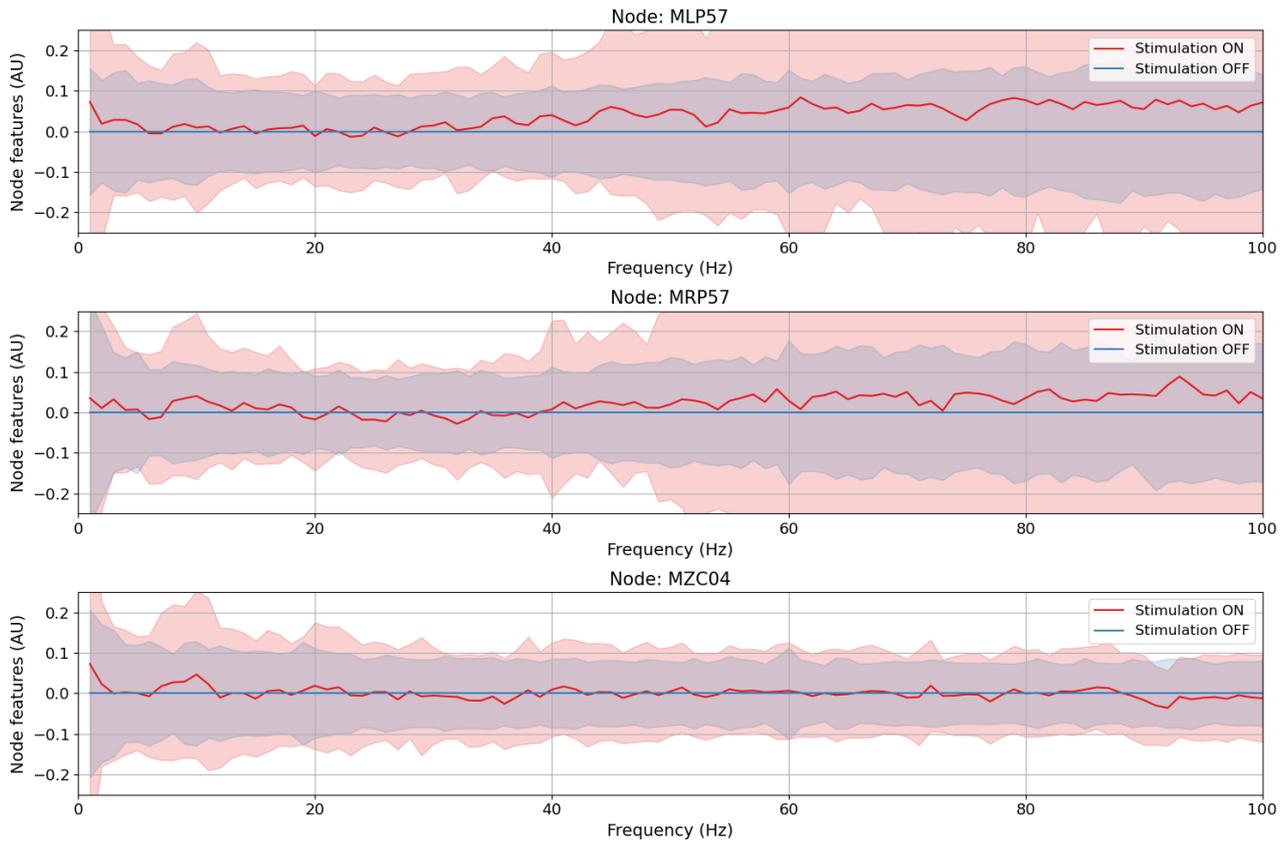
## 3.2. Graphs

### 3.2.1. Sensor Space Analysis

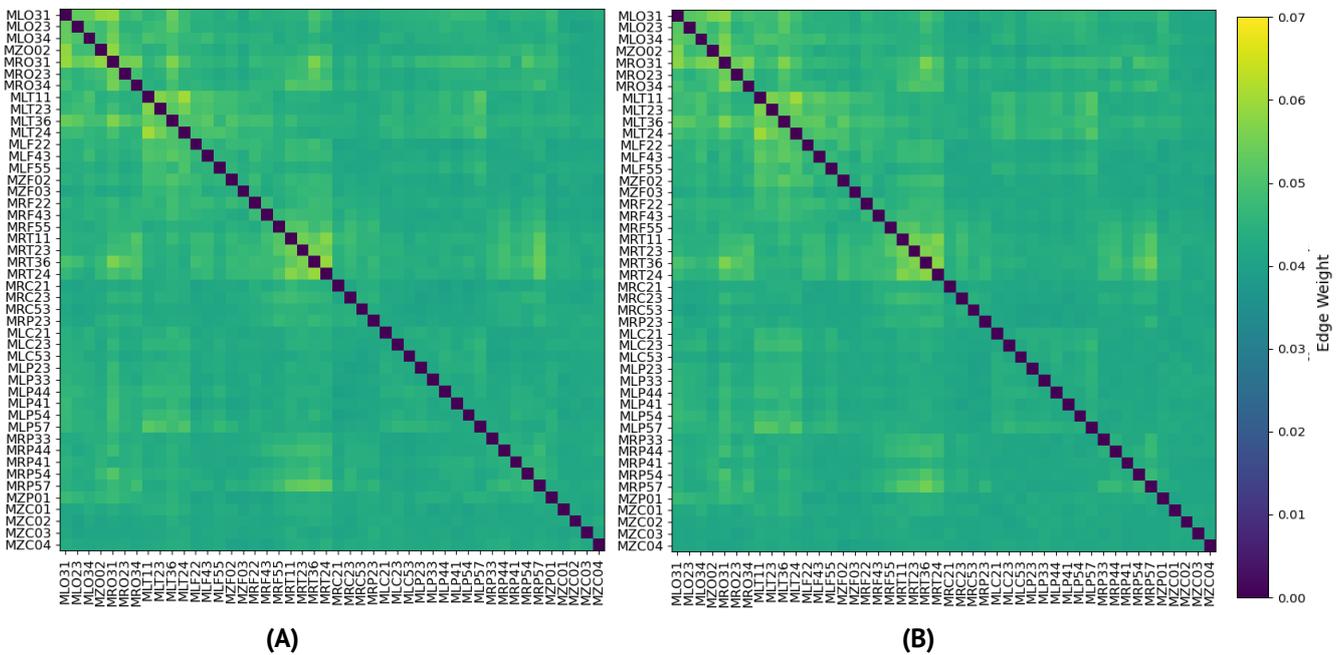
Using the sensor space analysis approach, graph attributes were computed based on MEG sensor data to construct graph-structured inputs for each frequency band of interest. A total of 950 subepochs across all files were created. Through the removal of artifacts, 20 subepochs were excluded, resulting in a total of 930 subepochs that were retained for graph construction. Among the resulting 930 graphs, 416 corresponded to stimulation OFF epochs, and 514 corresponded to stimulation ON epochs. The graph attributes for the full frequency band are presented in the following sections.

Figure 3.1 displays the average normalized node features for three nodes across all graphs separately for both stimulation conditions, with shading representing standard deviation. The three nodes represent sensors that reflect parietal areas, as well as a sensor in the midline that is roughly located around the central sulcus. The node features for all 46 nodes, averaged over all subepochs for stimulation OFF are shown in Figures L.3, and all node features averaged over all subepochs for stimulation ON are shown in Figures L.2.

Figure 3.2 presents the average PLI-based connectivity matrices for both stimulation states. The connectivity matrices show some patterns of connectivity, including some stronger connections within both left and right temporal lobes (MEG sensors MRT11, MRT23, MRT36, and MLT11, MLT23, MLT36), as well as connections within the occipital lobes (corresponding to the MEG sensors MLO31, MRO31, MZO02). Furthermore, increased connectivity was demonstrated by sensor MRP57 with temporal sensors MRT11, MRT23, MRT36, and MRT24. The difference in connectivity between the stimulation ON and OFF states is visualized in Figure L.1. This figure reveals an increase in connectivity during stimulation ON within frontotemporal areas in the left hemisphere (MLT11, MLC21, MLC23, and MLF43), as well as increased connectivity captured between frontotemporal sensor MLT11 and sensors located over occipital areas (MLO31, MZO02, and MRO31). Increased connectivity during stimulation OFF was primarily focussed



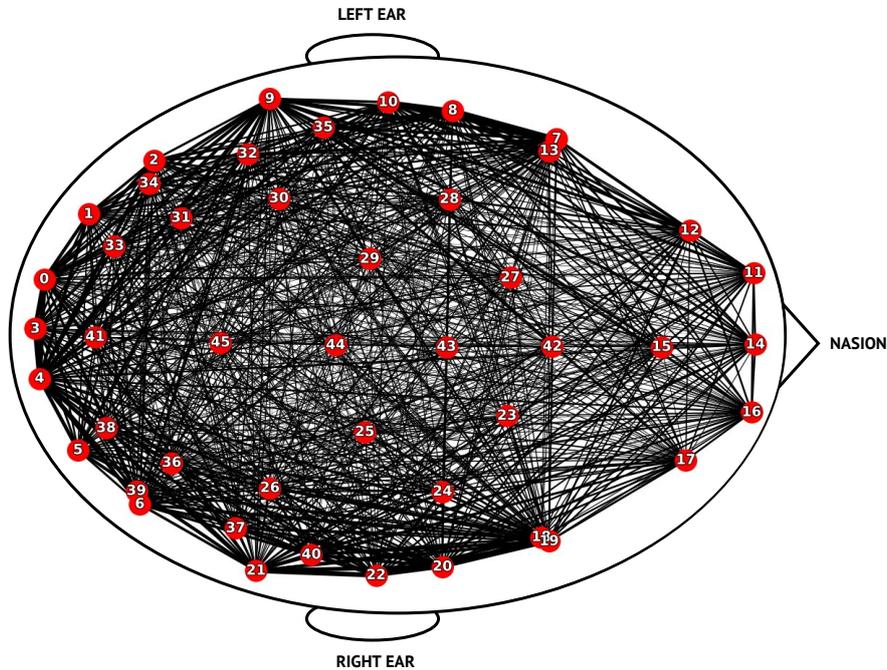
**Figure 3.1:** The average node features for three nodes (sensors corresponding to left and right parietal areas (MLP57 and MRP57) and a sensor in the midline at the central sulcus (MZO04)), averaged over all stimulation ON (red) and OFF (blue) subepochs. The standard deviation is shown as shading. AU: Arbitrary Units.



**Figure 3.2:** Connectivity matrices computed using the Phase Lag Index (PLI), representing the edges of the full frequency (1 - 100 Hz) sensor-based graph, averaged over all subepochs during stimulation OFF (A) and stimulation ON (B). Appendix G shows an overview of the selected nodes and the node labels.

between left and right parietal sensors (MLP41, MLP54, MLP57, MRP41, MRP54, and MRP57), and within parietal and central sensors (MRC53, MRP23, MLC21, MLC53, MZC03, MZC02).

The resulting averaged sensor-based graph is visualized in Figure 3.3, where the relative edge weights are illustrated through line thickness.



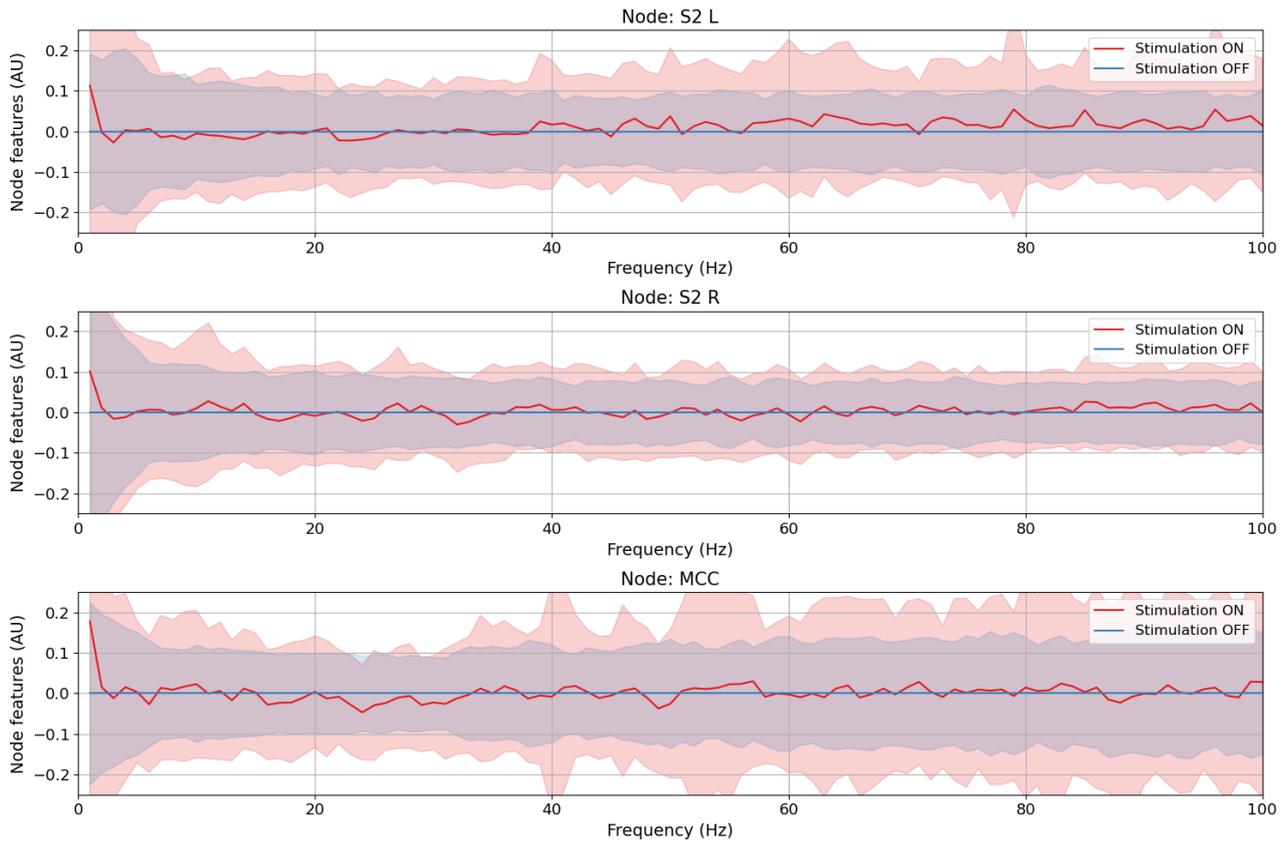
**Figure 3.3:** The resulting sensor-based graph visualized on a schematic head model, averaged across all graphs. See Appendix G for the sensor indices.

### 3.2.2. Source Space Analysis

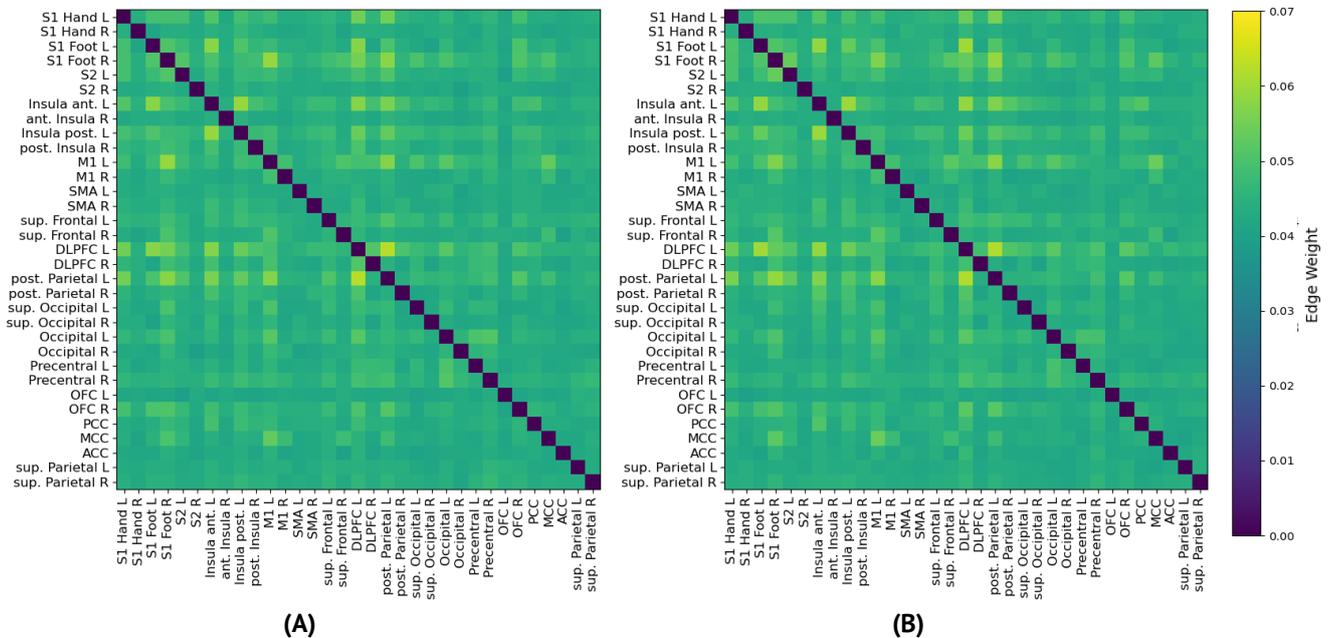
Of the 950 subepochs that were created for the source-based dataset with 35 MEG recordings, 35 subepochs were removed based on the presence of noise and artifacts, resulting in a total of 915 subepochs for the construction of the source-based graphs. Of these, 506 graphs had the label stimulation ON, and 409 graphs had the label stimulation OFF.

Figure 3.4 shows the average normalized node features across all subepochs in three nodes, e.g. cortical brain regions, for both stimulation states separately. These three brain regions represent the secondary somatosensory cortex (S2) of the left and right hemisphere, and the middle cingulate cortex (MCC). An overview of the average node features for all brain regions included in the graph is included in Figures L.5 and L.6, for the subepochs with stimulation OFF and ON, respectively.

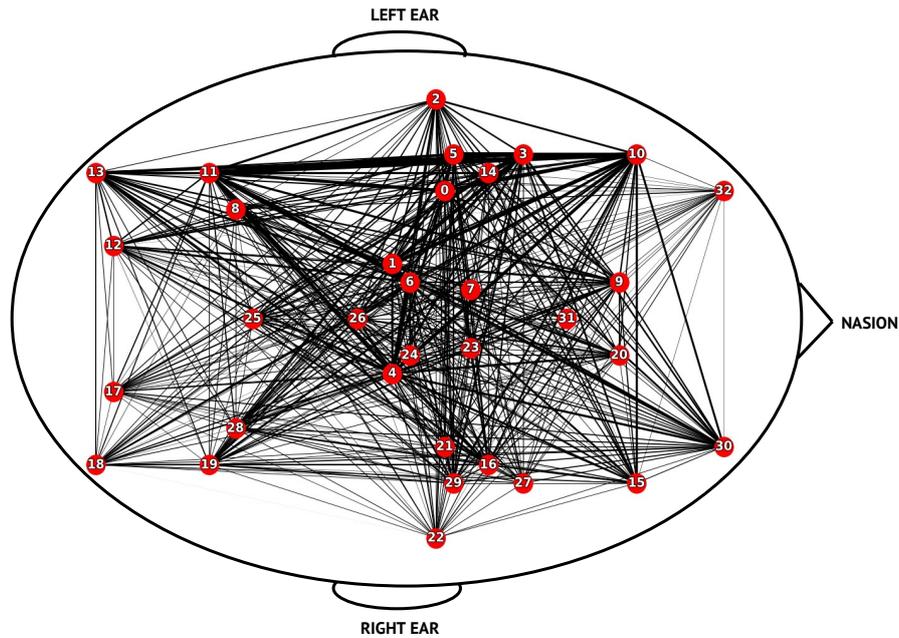
The average connectivity matrices across all subepochs for both stimulation states are presented in Figure 3.5. The connectivity matrices showed some patterns of increased connectivity related to the left dorsolateral prefrontal cortex, left posterior parietal lobule, bilateral anterior insula, left posterior insula, and the left primary somatosensory cortex. The difference in connectivity matrices between the stimulation ON and OFF states is shown in Figure L.4. During stimulation ON, increased connectivity was observed within the somatosensory cortices (between left and right primary somatosensory cortex (S1) and left secondary somatosensory cortex (S2)), as well as between the primary motor cortex and the superior frontal cortex. In contrast, during stimulation OFF, the right parietal cortex exhibited stronger connectivity with multiple brain regions. Additionally, the occipital cortices demonstrated increased connectivity with motor cortices during stimulation OFF.



**Figure 3.4:** The average node features for three nodes (left and right secondary somatosensory cortex (S2), and middle cingulate cortex (MCC)), averaged over all stimulation ON (red) and OFF (blue) subepochs. The standard deviation is shown as shading. AU: Arbitrary Units.



**Figure 3.5:** Connectivity matrices computed using the Phase Lag Index (PLI), representing the edges of the full frequency (1 - 100 Hz) source-based graph, averaged over all subepochs during stimulation OFF (A) and stimulation ON (B). Appendix D.1 shows an overview of the acronyms of the nodes in the source-space analysis.

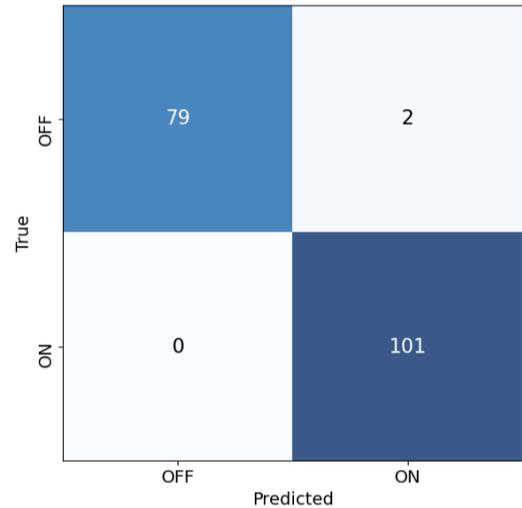


**Figure 3.6:** The resulting source-based graph visualized on a schematic head model, averaged across all graphs. See Appendix D.1 for source indices.

Figure 3.6 shows the average source-based graph, where edge thickness represents relative edge weights between nodes.

### 3.3. Model Outcomes

A graph classification architecture for the GNN model was set up with the aim of classifying active brain networks associated with stimulation activation and deactivation. Following the initialization of the GNN architecture, the model was trained and optimized on sensor-based graphs using the full frequency band (1 – 100 Hz). The same architecture was applied to other frequency bands to assess the differentiating characteristics of each band. Further experiments explored the classification performance of the GNN models across stimulation paradigms (tonic and burst) and recording sites (the MNI and the Donders institutes). Additionally, another GNN model was trained and evaluated using the source-based graphs with the full frequency band.



**Figure 3.7:** Confusion matrix of the test set for the sensor-based GNN model trained and tested on the full frequency band.

#### 3.3.1. Model Performance

##### 3.3.1.1. Sensor Space

Using the full frequency band, the sensor-based GNN model was able to successfully differentiate between stimulation ON and OFF graphs, with a validation and test accuracy of 0.99 and 0.98 (Table 3.1). The confusion matrix for the test set obtained from the GNN model for the full band (Figure 3.7) showed that almost all stimulation ON graphs were correctly classified (101/103), and all of the stimulation OFF graphs were correctly classified (79/79). Hence, zero false negatives and two false positives were outputted.

The beta and gamma bands showed validation and test accuracies of 0.97 and 0.98 (beta band) and

0.99 and 0.98 (gamma band), as shown in Table 3.1. Hence, these frequency bands yielded similar high performance as the full-band. The delta, theta, and alpha bands showed the lowest performance overall, with validation and test accuracies ranging from 0.76 to 0.83. Furthermore, the theta and alpha bands combined reached higher accuracies than the individual bands, resulting in a validation accuracy of 0.90 and a test accuracy of 0.92. Confusion matrices for the GNN models for the different frequency bands are shown in Appendix N.

An additional analysis was explored for the lower performing frequency bands (delta, theta, and alpha) by training and testing separate GNN models with an increased frequency resolution of 0.25 Hz. This resulted in a further decrease in classification performance for all three frequency bands, with test accuracies of 0.63, 0.75, and 0.65 for the delta, theta, and alpha bands, respectively.

**Table 3.1:** Validation and test accuracy for the sensor-based GNN models across different frequency bands.

Frequency band	Validation accuracy	Test accuracy	F1-score	AUC
<i>Full band</i>	0.99	0.99	0.99	0.99
<i>Delta band</i>	0.76	0.81	0.81	0.89
<i>Theta band</i>	0.80	0.78	0.78	0.85
<i>Alpha band</i>	0.77	0.83	0.83	0.89
<i>Theta &amp; alpha band</i>	0.90	0.92	0.92	0.98
<i>Beta band</i>	0.97	0.98	0.98	1.00
<i>Gamma band</i>	0.99	0.98	0.98	0.98

*AUC: Area Under the Curve.*

### 3.3.1.2. Tonic and Burst Stimulation

The GNN models for both stimulation paradigms revealed similar accuracies, namely validation accuracies of 0.97 and 0.98 and test accuracies of 0.98 and 0.97 for tonic and burst stimulation, respectively (Table 3.2). Confusion matrices for the GNN models for both stimulation paradigms are shown in Appendix N. Through testing the GNN model trained on tonic recordings using burst recordings, a cross-paradigm accuracy of 0.88 was reached (Table 3.3). However, a decreased performance was shown when testing the tonic recordings on the previously trained GNN model on burst recordings, showing an accuracy of 0.69.

**Table 3.2:** Validation and test accuracy for sensor-based GNN models trained and tested on MEG recordings from one stimulation paradigm.

Stimulation paradigm	Validation accuracy	Test accuracy	F1-score	AUC
<i>Tonic</i>	0.97	0.98	0.98	0.98
<i>Burst</i>	0.98	0.97	0.97	0.99

*AUC: Area Under the Curve.*

**Table 3.3:** Cross-paradigm performance for sensor-based GNN models trained on one stimulation paradigm and tested on the other stimulation paradigm.

Stimulation paradigm	Cross-paradigm accuracy	Cross-paradigm F1-score
<i>Tonic</i>	0.88	0.88
<i>Burst</i>	0.69	0.69

### 3.3.1.3. Recording Institutes

The model trained and tested on data from the MNI performed slightly better than the model trained and tested on data from the Donders Institute, while both showing high performance (0.99 vs. 0.97 for the validation accuracy and 0.99 vs. 0.98 for the test accuracy, as shown in Table 3.4). Confusion matrices for the GNN models for both recording institutes are shown in Appendix N. In line with this, the cross-site performance of the model trained on the MNI recordings was higher compared to the cross-site performance score for the model trained on Donders Institute data (0.87 vs. 0.81 for the respective test accuracies, shown in Table 3.5).

**Table 3.4:** Validation and test accuracy for sensor-based GNN models trained and tested on MEG recordings from each of the recordings institutes.

Recording institute	Validation accuracy	Test accuracy	F1-score	AUC
<i>MNI</i>	0.99	0.99	0.99	1.00
<i>Donders</i>	0.97	0.98	0.98	1.00

*AUC: Area Under the Curve; MNI: Montreal Neurological Institute.*

**Table 3.5:** Cross-site performance for sensor-based GNN models trained on data from one recording institute and tested on data from the other institute.

Recording institute	Cross-site accuracy	Cross-site F1-score
<i>MNI</i>	0.87	0.87
<i>Donders</i>	0.81	0.81

*MNI: Montreal Neurological Institute.*

**Table 3.6:** Validation and test accuracy for the source-based GNN model.

Validation accuracy	Test accuracy	F1-score	AUC
0.99	0.97	0.97	1.00

*AUC: Area Under the Curve*

### 3.3.1.4. Source Space

The source-based GNN model reached a validation accuracy of 0.99 and a test accuracy of 0.97, thereby showing similar performance to the sensor-based GNN model for the full frequency band (Table 3.6). All graphs with the label stimulation ON were correctly classified by the sensor-based GNN model (101/101), and a large amount of the stimulation OFF graphs were correctly classified (77/82), as shown in the confusion matrix (Figure 3.8). This resulted in five false positives and zero false negatives.

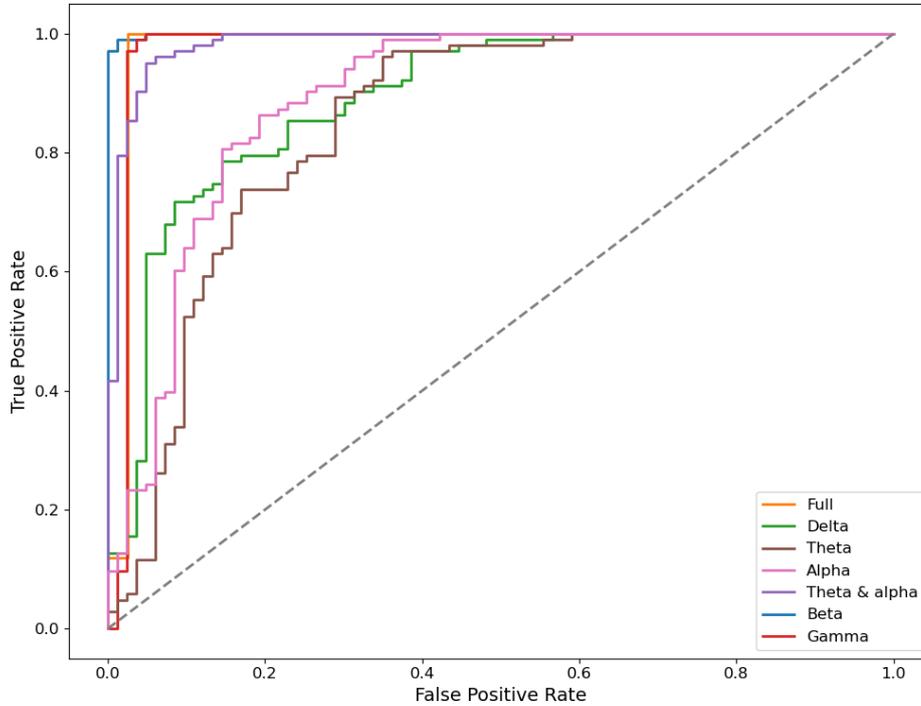
### 3.3.2. Receiver Operating Characteristic Curves

Figure 3.9 shows the ROC curves for the sensor-based GNN models trained on different frequency bands. The ROC curves indicate the highest performance in the full, gamma, and beta frequency bands. In contrast, the delta, theta, and alpha bands

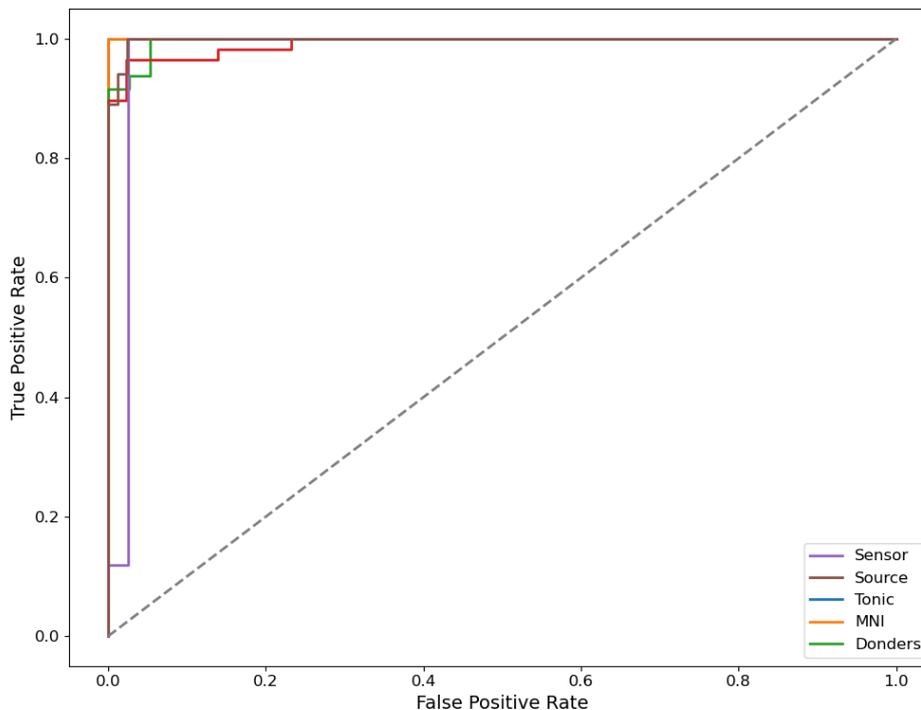
True	Predicted	
	OFF	ON
OFF	77	5
ON	0	101

**Figure 3.8:** Confusion matrix of the test set for the source-based GNN model trained and tested on the full frequency band.

individually show reduced classification accuracy. However, when the theta and alpha bands are combined, performance improves compared to using either band alone. Figure 3.10 presents a comparison of ROC curves for the remaining sensor-based model configurations as well as the source-based model. These ROC curves demonstrate that all full-band GNN model configurations, across both stimulation paradigms, both recording institutes, and including the source-based GNN model, consistently exhibit similarly high classification performance.



**Figure 3.9:** ROC curves for sensor-based GNN models trained on the different frequency bands.



**Figure 3.10:** ROC curves for sensor-based GNN models trained on the full-frequency band, the sensor-based GNN models trained on different stimulation paradigms (tonic and burst), different recording institutes (the Montreal Neurological Institute (MNI) and the Donders Institute), and the source-based GNN model.

### 3.3.3. Hyperparameter Settings

The hyperparameter settings that yielded the highest classification performance for each GNN model configuration are summarized in Table 3.7. While optimal settings varied across models, several consistent patterns emerged. A batch size of at least eight was selected in all configurations, and smaller batch sizes (two or four) were not among the best-performing configurations. The number of hidden channels was most frequently set to the maximum value of 128. Most models used either three or four convolutional layers, although two layers were optimal in three configurations. The dropout rate was most often set to the minimum value of 0.01, with higher dropout rates of 0.3 or 0.5 never selected. A learning rate of 0.001 was the most commonly used, though a few models performed well with 0.01 or 0.0001 as learning rate. The best-performing model configurations did not include a learning rate of 0.00001. Furthermore, most models applied some form of edge filtering.

**Table 3.7:** Outcomes for the hyperparameters for each GNN model.

Sensor-based model	Batch size	Hidden channels	Dropout rate	Learning rate	Layers	Threshold	Top-K
<i>Full band</i>	64	128	0.01	0.001	2	0.05	None
<i>Delta band</i>	128	128	0.1	0.001	3	None	600
<i>Theta band</i>	16	128	0.1	0.01	3	0.01	600
<i>Alpha band</i>	8	128	0.01	0.001	3	0.07	None
<i>Theta &amp; alpha band</i>	128	128	0.1	0.001	4	None	300
<i>Beta band</i>	8	128	0.01	0.0001	4	None	300
<i>Gamma band</i>	64	32	0.01	0.001	4	0.05	900
<i>Tonic stimulation</i>	16	64	0.01	0.001	4	0.07	600
<i>Burst stimulation</i>	128	16	0.5	0.01	3	None	None
<i>MNI</i>	16	64	0.01	0.0001	2	None	300
<i>Donders Institute</i>	8	64	0.01	0.01	4	0.05	300
Source-based model	Batch size	Hidden channels	Dropout rate	Learning rate	Layers	Threshold	Top-K
<i>Full band</i>	64	128	0.1	0.01	3	0.05	None

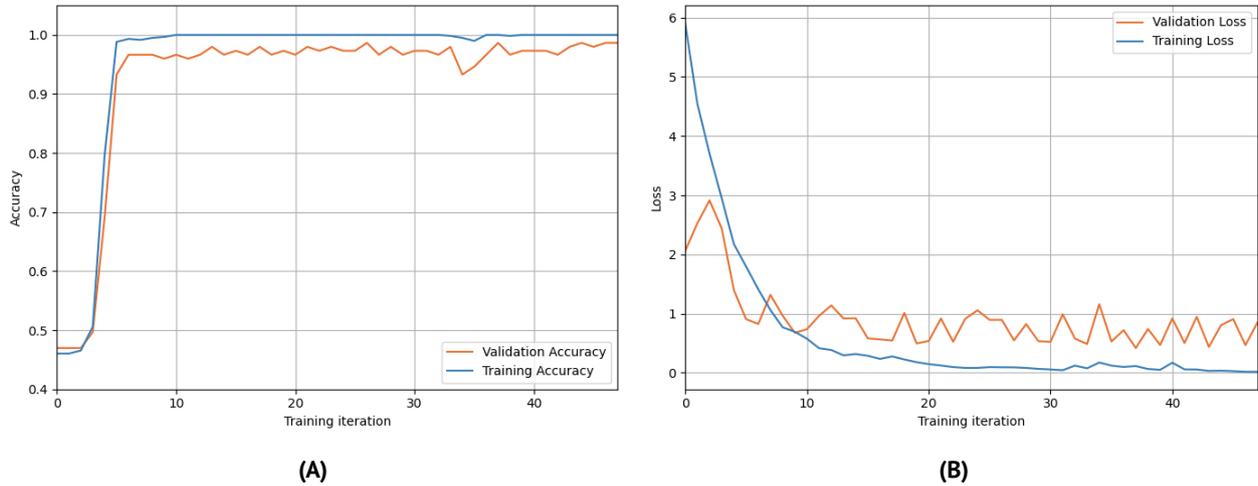
### 3.3.4. Training and Validation Accuracy and Loss

The performance of both the GNN models was evaluated by tracking their training and validation accuracy and loss over the course of training, using the best-performing hyperparameter configurations. Training and validation accuracy curves exhibited a rapid improvement early in training in both the sensor-based GNN (Figure 3.11A) and source-based GNN (Figure 3.12A). After this initial rise, the training and validation accuracy plateaued with minimal further gains. The close alignment between training and validation accuracy throughout the training process supports the stability and generalizability of both models.

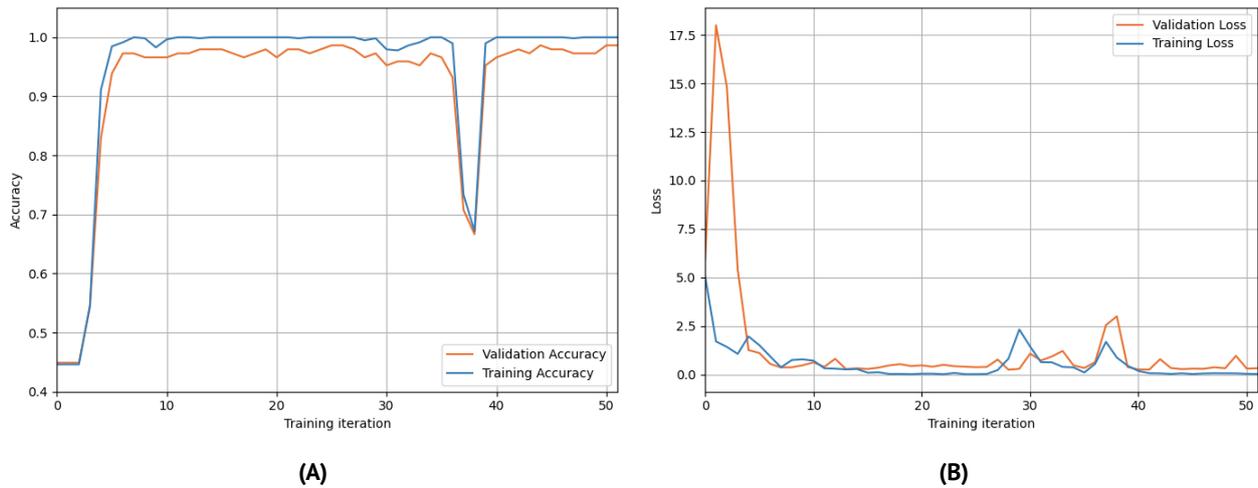
For the sensor-based GNN model, a brief decline in validation accuracy occurred around iteration 34, followed by a recovery and continued upward trend. In contrast, the source-based GNN model experienced a more pronounced drop in both training and validation accuracy at iteration 37. However, this decline was short-lived and within three additional iterations, the model rebounded and stabilized, achieving high and sustained accuracy levels in subsequent training iterations.

When examining the loss curves, the sensor-based GNN model demonstrated a clear and steady decrease in training loss (Figure 3.11B), approaching near-zero values. The validation loss showed a more gradual decline and more fluctuations throughout model training in the sensor-based GNN model. The source-

based GNN model, on the other hand, achieved overall lower loss values for both training and validation compared to the sensor-based model (Figure 3.12B). Nevertheless, the training process began with a sharp spike in validation loss, followed by a substantial drop that brought validation loss closely in line with the training loss. Notably, spikes in loss at iterations 29 and 37 coincided with temporary dips in the accuracy of the source-based GNN model.



**Figure 3.11:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model for the full frequency band.



**Figure 3.12:** Validation and test accuracy (A) and loss (B) over the training iterations for the source-based GNN model for the full frequency band.

The accuracy and loss curves for all other GNN models that were trained in this study are presented in Appendix M.

### 3.4. Node Explainability

Node explainability was implemented using two recordings (tonic and burst stimulation) on both the sensor and source-based GNN models for the full frequency band. For each recording, the overall most important nodes were retrieved, along with the average fidelity and sparsity scores across all graphs. The tonic recording included a total of 50 graphs and the burst recording included a total of 25 graphs.

### 3.4.1. Sensor Space

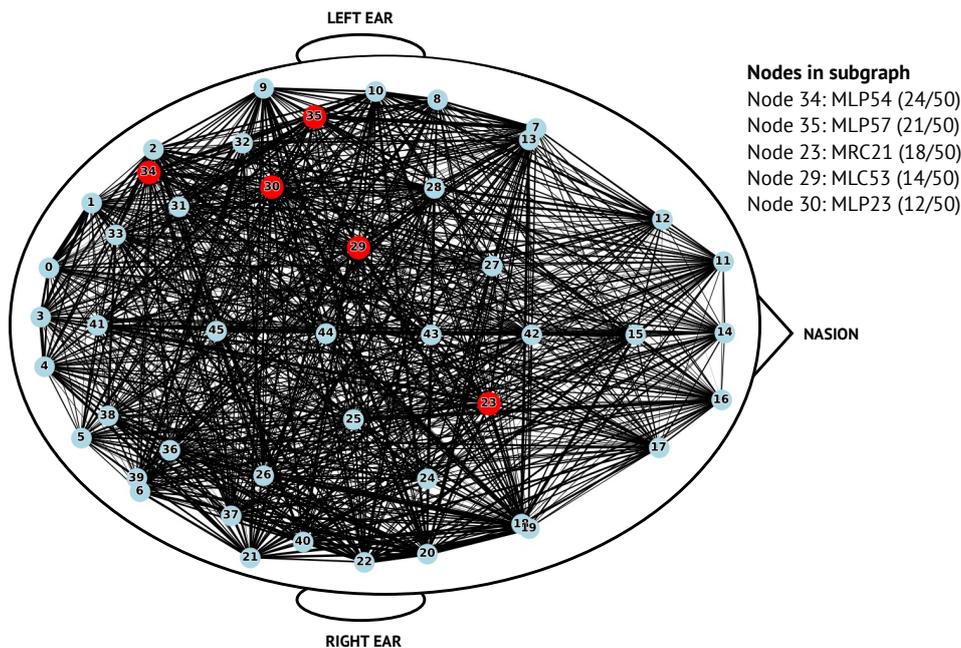
For the sensor-based GNN model, the tonic and burst stimulation recordings were run separately through the explainability module. The resulting fidelity and sparsity scores are reported in Table 3.8.

The tonic recording showed an average fidelity score of 0.70, with a sparsity score of 0.89. The nodes that were overall most frequently identified as the subgraph in this recording were the following MEG channels: MLP54, MLP57, MRC21, MLC53, and MLP23. Of these, MLP54 and MLP57 were included in most important nodes in 24 and 21 of the total 50 graphs for this tonic recording, respectively. These nodes are highlighted in Figure 3.13.

**Table 3.8:** Node explainability results on the sensor-based GNN model using tonic and burst recordings from *PT06*, showing the overall most important nodes, the average fidelity scores, and the sparsity scores.

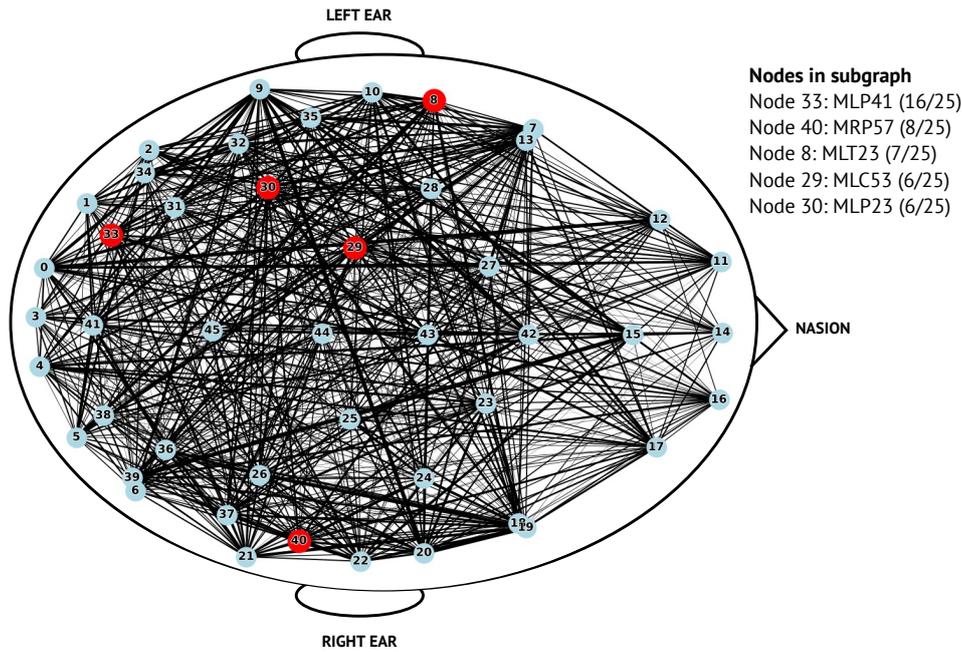
MEG recording	Overall most important nodes	Average fidelity score	Sparsity score
<i>Tonic</i>	MLP54, MLP57, MRC21, MLC53, MLP23	0.70	0.89
<i>Burst</i>	MLP41, MRP57, MLT23, MLC53, MLP23	0.75	0.89

MEG: Magnetoencephalography



**Figure 3.13:** Graph-based representation of *PT06* (tonic stimulation) showing the identified subgraph, using the sensor-based GNN. The count of each highlighted node in the subgraph relative to the total number of graphs inside this recording is included. *Appendix G* shows an overview of the selected nodes and indices.

An average fidelity score of 0.75 was computed based on the identified subgraph in the burst recording. The sparsity score of 0.89 was identical to the tonic recording, as the number of nodes in the identified subgraph relative to the total number of graphs stayed the same. The overall subgraph that was most frequently identified as the subgraph over all graphs in this recording were the nodes: MLP41, MRP57, MLT23, MLC53, and MLP23. This subgraph is highlighted in Figure 3.14. The node MLP41 was highlighted in the majority of the graphs created based on the burst recording (16/25).



**Figure 3.14:** Graph-based representation of *PT06* (burst stimulation) showing the identified, using the sensor-based GNN. The count of each highlighted node in the subgraph relative to the total number of graphs inside this recording is included. *Appendix G* shows an overview of the selected nodes and indices.

### 3.4.2. Source Space

In the source-based GNN model, explainability results for both tonic and burst stimulation recordings were retrieved, as shown in Table 3.9.

**Table 3.9:** Node explainability results on the source-based GNN model using tonic and burst recordings from *PT06*, showing the overall most important nodes, the average fidelity score, and the sparsity score. *Appendix D.1* shows an overview of the nodes included the source-space analysis.

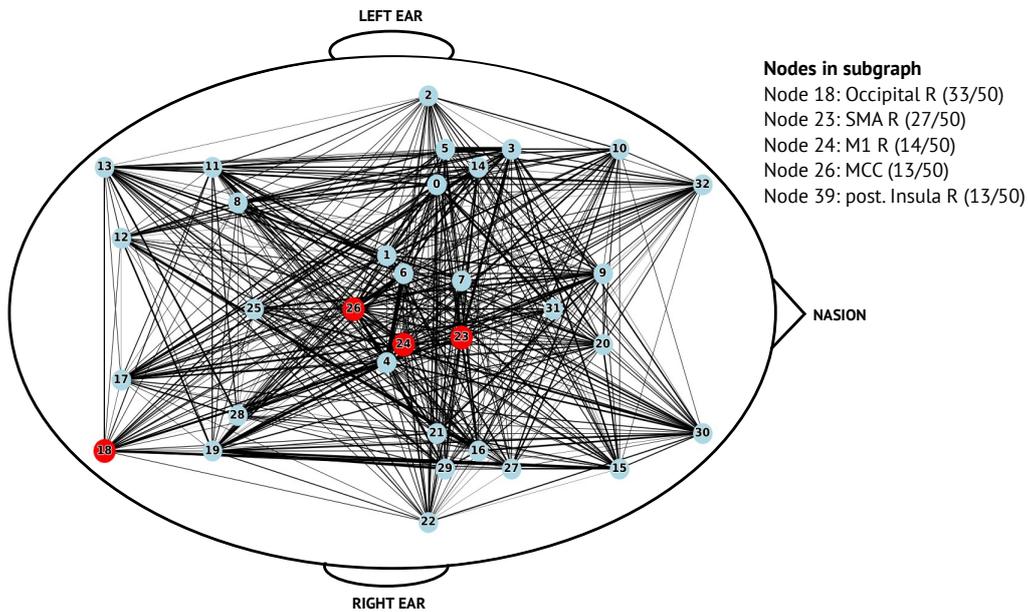
MEG recording	Overall most important nodes	Average fidelity score	Sparsity score
<i>Tonic</i>	Occipital lobe R, SMA R, M1 R, MCC, posterior insula R	0.86	0.85
<i>Burst</i>	Sup. frontal lobe L, occipital lobe L, anterior insula R, sup. parietal lobe R, ACC	0.87	0.85

ACC: Anterior Cingulate Cortex; MCC: Middle Cingulate Cortex; M1: primary motor cortex; MEG: Magnetoencephalography; R/L: right/left; Sup.: superior.

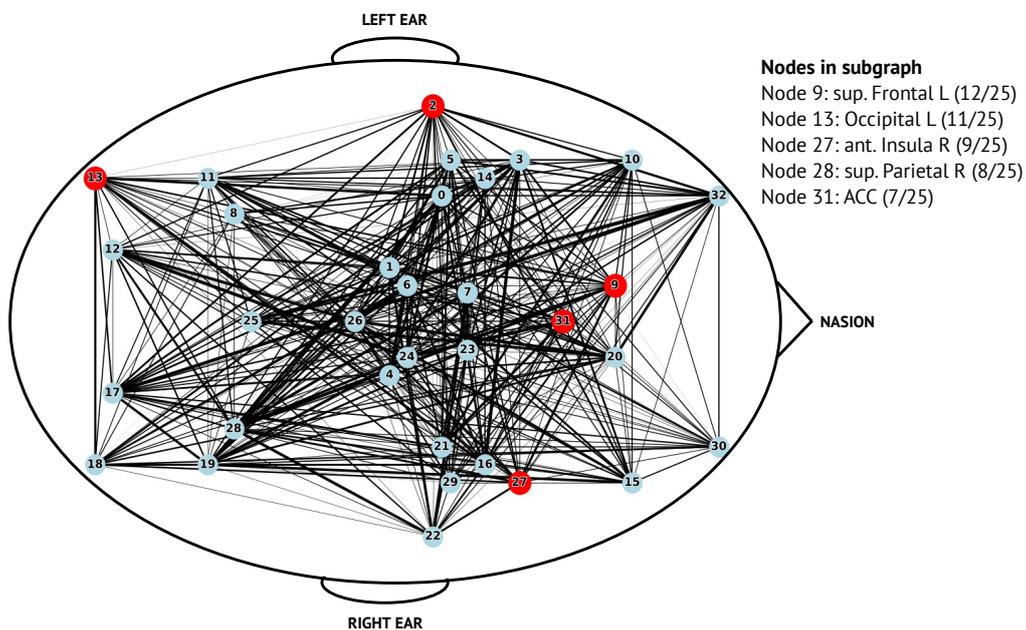
The tonic recording showed an average fidelity score of 0.86 and a sparsity score of 0.85. From all 40 graphs created for the tonic stimulation recording, the most important nodes that were identified included the right occipital lobe, right supplementary motor area (SMA), right primary motor cortex (M1), middle cingulate cortex (MCC), and the right posterior insular cortex. Of these, the right occipital lobe and the right SMA were identified in over half of the graphs (in 33/50 and 27/50 graphs). The identified subgraph is highlighted in Figure 3.15.

The average fidelity score and sparsity score found for the burst recording were 0.87 and 0.85, respectively. The left superior frontal lobe, left occipital lobe, right anterior insular cortex, right superior parietal lobe, and the anterior cingulate cortex (ACC) were overall most identified as important nodes in the burst recording. The left superior frontal lobe and the left occipital lobe had the largest count relative to the

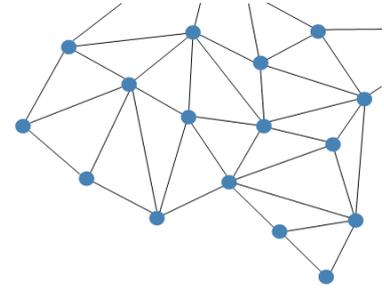
total number of graphs (12/25 and 11/25, respectively). The overall most important nodes are shown in Figure 3.16.



**Figure 3.15:** Graph-based representation of *PT06* (tonic stimulation) showing the identified subgraph, using the source-based GNN. The count of each highlighted node in the subgraph relative to the total number of graphs inside this recording is included. *Appendix D.1* shows an overview of the nodes included in the source-space analysis.



**Figure 3.16:** Graph-based representation of *PT06* (burst stimulation) showing the identified subgraph, using the source-based GNN. The count of each highlighted node in the subgraph relative to the total number of graphs inside this recording is included. *Appendix D.1* shows an overview of the nodes included in the source-space analysis.



## 4 | Discussion

In this study, I successfully developed Graph Neural Network (GNN) models to classify MEG recordings of chronic pain patients treated with Spinal Cord Stimulation (SCS) using cyclic stimulation settings, providing an analysis framework for further research. More specifically, graph-based representations of the resting-state Magnetoencephalography (MEG) recordings were created, a model architecture for graph classification was developed, and the model performance for different inputs was evaluated. Furthermore, node explainability methods were incorporated to explore how the GNN model decided on a classification, revealing active brain networks associated with stimulation.

### 4.1. Graph Dataset

Graph datasets were constructed using either MEG channels or source-reconstructed brain regions as nodes. Node features were derived from baseline-corrected Power Spectral Density (PSD) values, while edges represented functional connectivity computed via the Phase Lag Index (PLI) metric. These graphs were constructed separately for each frequency band, as well as for each stimulation paradigm and recording institute, providing a structured representation of brain activity and connectivity across these different configurations.

This graph construction approach aligns with prior neuroimaging-based graph studies [58, 59, 106, 107]. However, the literature does not provide a consensus on optimal graph definitions, including nodes, node features, and edges, or processing pipelines for brain connectivity analysis. The effectiveness of any specific method is context-dependent and subject to assumptions embedded in the graph design [30, 108].

#### 4.1.1. Node Features

The use of the PSD as node feature provides a representation of frequency-specific neural activity at every node. Compared to time-series data or statistical metrics, PSD-based node features are lower-dimensional, making them more suitable for graph-based representations of MEG recordings.

However, raw PSD magnitudes varied substantially between recordings, likely due to MEG-specific factors such as sensor offsets and slow ambient field drifts, or anatomical factors such as the thickness of the skull, or placement of the patient's head relative to the MEG sensors. To mitigate this, a normalization step was implemented to align feature magnitudes across recordings. As no true baseline was available, given that both stimulation ON and OFF states are of interest, the PSD from stimulation OFF subepochs was used as a pseudo-baseline. This normalization approach involved subtracting the mean PSD from stimulation OFF subepochs and then dividing by this mean. As a result, node features from stimulation OFF graphs centred around zero with small variability, while node features from stimulation ON graphs reflected relative deviations from this baseline. While this normalization strategy reduced variance and facilitated training across the entire dataset, it limited the interpretability of the node feature values, as the node feature representations shifted from absolute frequency power to relative changes in frequency power between stimulation states. While the stimulation OFF condition serves as a pseudo-baseline, it may still reflect lingering effects from prior stimulation or other dynamic changes that occur due to chronic pain. Given that the carry-over period of SCS is not completely understood and likely varies considerably between

individuals, obtaining a true baseline remains challenging [65]. An alternative normalization approach involves calculating the relative power of each frequency band as a proportion of the total power. This method is particularly useful when differences across stimulation ON and OFF states are concentrated in specific frequency ranges, as it emphasizes the relative contribution of each band while minimizing inter-patient amplitude variability.

#### 4.1.2. Edges

Edges in the graphs were defined by computing pairwise PLI values between nodes, resulting in undirected, weighted graphs. PLI is particularly well-suited for this application, as it reduces the influence of volume conduction and captures consistent phase relationships between brain regions over time [87]. Even though the effect of volume conduction in MEG recordings may not be as pronounced as in Electroencephalography (EEG), spurious zero-lag correlations can still occur due to field spread and signal leakage. By emphasizing non-zero phase lag interactions, PLI enhances the specificity of the connectivity estimates, thereby providing a more physiologically meaningful representation of functional brain networks. This makes PLI an appropriate choice for constructing graphs that aim to reflect true interregional communication rather than artifacts of signal mixing. However, it is important to note that PLI exclusively reflects phase coupling and does not account for amplitude-based interactions, potentially limiting the interpretability of the full spectrum of brain connectivity. Thus, while the PLI-based connectivity offers robustness against volume conduction, it provides only a partial view of the complex neural interactions underlying chronic pain and SCS effects [91].

From the sensor space analysis, a few patterns of increased connectivity were found within the temporal lobes and occipital lobes. Moreover, increased connectivity was observed between temporal brain regions and the right parietal lobe. Differences in connectivity between stimulation ON and OFF states revealed that stimulation OFF was associated with enhanced connectivity in central and parietal regions, approximately corresponding to sensorimotor areas. These changes may reflect alterations in pain perception, as turning stimulation OFF can lead to the re-emergence of pain, potentially increasing engagement of pain-related networks. However, the exact onset of pain re-emergence for each patient is not precisely known [65]. During stimulation ON, increased connectivity was found in left frontotemporal areas, potentially capturing brain activity in the insula. This finding may reflect modulation of the medial pain pathway by SCS [12].

The source space analysis revealed more widespread patterns of connectivity across different brain regions, likely due to the ability of source reconstruction methods to estimate neural activity directly at the cortical level. By resolving overlapping signals recorded at spatially distant sensors and projecting them onto anatomically distinct brain areas, a more precise identification of interactions between distributed brain regions can be achieved, thereby uncovering more distributed connectivity patterns that may remain obscured in sensor-level analyses.

In the source space analysis, elevated PLI-based connectivity was observed between the dorsolateral prefrontal cortex (DLPFC) and the left posterior parietal lobule. Both of these regions also showed strong connections with the bilateral anterior insula, the left posterior insula, and the left primary somatosensory cortex (S1). Furthermore, the left primary motor cortex (M1) exhibited increased connectivity with the somatosensory cortex, particularly in the foot representation area. These patterns are consistent with activation of the ascending lateral pain pathway, which is implicated in the sensory-discriminative aspect of pain, potentially indicating the experience of pain [23]. Additionally, the insular cortex is associated with the affective-motivational processing of pain [109, 110]. Differences in connectivity between stimulation ON and OFF states highlighted increased connectivity within somatosensory cortices during stimulation ON, while stimulation OFF states showed increased connectivity in widespread networks including the

right superior parietal cortex, the left anterior insula, and right secondary somatosensory cortex (S2). While these findings indicate a potential trend in altered brain connectivity between the ON and OFF conditions, further research with statistical analysis is required to confirm these effects and to elucidate how SCS modulates specific functional brain networks.

To date, only a limited number of studies have investigated the effects of SCS on resting-state functional connectivity [32, 111, 112]. One previous study by Deogaonkar et al. evaluated functional connectivity using functional Magnetic Resonance Imaging (fMRI) during stimulation ON and OFF conditions, with each state lasting ten minutes, and reported stronger connectivity between frontal, sensorimotor, and cingulate cortices during stimulation ON, and increased connectivity between limbic and somatosensory areas during stimulation OFF [111]. These findings are only partially consistent with the results of the present study. Several factors may account for this discrepancy. First, the persistent presence of chronic pain in some patients, even during stimulation ON, may cause variability in stimulation-related connectivity changes. In contrast, the study by Deogaonkar et al. included only patients who experienced more than 50% pain reduction following SCS, potentially making these stimulation effects more pronounced. Furthermore, the potential presence of stimulation carry-over effects, where the neurophysiological impact of stimulation persists after the stimulation is turned OFF, may obscure acute differences between ON and OFF states [65]. The study by Deogaonkar et al. attempted to control for this by excluding patients with carry-over effects exceeding the ten-minute stimulation cycle, and hence, this approach only reflects early responders and may not generalize to the broader patient population.

These factors highlight the complexity of interpreting functional connectivity changes in the context of SCS. Therefore, future studies may benefit from more individualized analyses, accounting for patient-specific stimulation effects and timing these effects, or from study designs that include patient groups that are matched on their washout periods of SCS to minimize carry-over effects in group-level analysis.

### 4.1.3. Frequency Band Graphs

Graph construction was performed separately for distinct frequency bands using the sensor space analysis, allowing the model to learn frequency-specific patterns within the graph attributes. This approach aligns with prior findings that brain networks display distinct organizational structures across different frequency bands [80, 81]. By providing a more detailed view of brain dynamics, frequency-specific graphs may help identify stimulation-related changes and highlight frequency-dependent contributions of individual brain regions. This supports the implementation of node-level explainability analyses in future studies, aimed at uncovering which node features, i.e. which frequency bins, are most influential for classification.

### 4.1.4. Class Imbalance

A class imbalance was present in the graph datasets, with more stimulation ON than stimulation OFF graphs. This was caused by the cyclic structure of the recordings and the method used to segment data into subepochs. As most recordings ended mid-cycle, fewer OFF-labelled segments were captured. Future MEG recordings during cyclic stimulation should include the same amount of cycles for both stimulation states, since balancing the classes in this research is essential to ensure that the model learns representative features from both stimulation ON and OFF conditions. This can potentially increase classification performance and may enable the extraction of clinically meaningful insights by identifying neural patterns associated with both stimulation ON and OFF states, rather than differentiating between them.

## 4.2. Graph Neural Network Model

A GNN model architecture was developed with the aim of learning and distinguishing the brain network patterns associated with SCS. Specifically, a graph classification framework was implemented to process and classify graphs derived from cyclic SCS recordings, where each graph represents features from MEG data during either stimulation ON or OFF states. The model was initially developed and optimized using sensor space data, where nodes correspond to MEG sensors and both functional connectivity and node features were computed across the full frequency range (1–100 Hz). This approach enabled the model to capture the broad spectrum of neural dynamics associated with SCS. After model training and optimization, the GNN demonstrated a strong ability to distinguish between stimulation ON and OFF conditions, indicating that the extracted graph-based features effectively reflect the underlying changes in brain network activity driven by SCS.

## 4.3. Model Performance

The developed GNN model architecture was implemented for a variety of different input configurations, including different frequency bands, different stimulation paradigms, different recording locations, and additionally, using source-based graphs.

### 4.3.1. Frequency Bands

Across all frequency bands, the models trained on the full band, as well as the beta and gamma bands, demonstrated the highest classification performance, suggesting that higher frequency activity carries critical information for differentiating between stimulation ON and OFF conditions. In contrast, delta, theta, and alpha bands showed relatively lower performance, possibly due to the presence of less discriminative features within these frequency ranges. These findings are partly supported by literature, since previous studies described that the alpha and gamma bands are mostly important in pain processing and neuromodulation [113]. Kim et al. [114] describe that the alpha band is responsible for pain sensitivity and changes in the pain character, while the gamma band is more focussed on the conscious perception and attentional processing of pain. Therefore, the results of the current study suggest that pain sensitivity is less affected through SCS, however, the conscious perception of pain is altered during stimulation ON states. Furthermore, abnormal beta band power has been reported in chronic pain patients, where the reduced power in the beta band might be reflective of sensory processing in response to pain [115]. This finding is supported by the outcomes of the current study, indicating that the beta band was highly discriminative for classification SCS states.

Interestingly, combining theta and alpha bands improved classification performance over either band alone. This finding may be attributable to the increased frequency range and resolution, enabling the model to extract more and richer features. However, despite this, an increased frequency resolution of 0.25 Hz, compared to the initial 1 Hz, was explored by training and testing separate GNN models that were trained and tested using the smallest frequency bands (delta, theta, and alpha), resulting in decreased classification performance for all three frequency bands. This suggests that the information in the lower frequency bands might not contain as many discriminative features, compared to the higher frequency bands, and through an increase in frequency bins, this could not be improved. Another possible explanation is that the initial GNN architecture was optimized for the full frequency range and therefore, it may not be well-suited to effectively learn patterns from narrower frequency bands, even when provided with higher frequency resolution.

Furthermore, because stimulation artifacts were predominantly present within the gamma frequency

range, and the applied filtering steps may not have completely eliminated these artifacts, it is possible that residual stimulation-related artifacts contributed to the elevated classification performance observed in the gamma band, and consequently also in the full band. In contrast, the beta band, where no stimulation-specific filtering was applied, also showed high performance, suggesting that its discriminative power is less likely to be driven by artifacts. To better understand the extent of residual artifacts, a follow-up approach could involve training a GNN model on MEG recordings in which the SCS was set to a fixed stimulation frequency, ensuring consistent artifact removal through filtering, and subsequently testing this model on MEG data acquired with different stimulation frequencies. This would help evaluate whether the model's performance is influenced by stimulation frequency-specific residual signals rather than genuine neurophysiological differences.

#### 4.3.2. Stimulation Paradigms

Classification performance differences between tonic and burst stimulation recordings were generally minimal, indicating that both paradigms contain distinct features associated with the stimulation ON and OFF states. However, when evaluating across paradigms, more pronounced differences in classification performance were observed. Specifically, a GNN model trained on tonic SCS recordings showed only a modest decrease in accuracy when tested on burst SCS data, while a model trained on burst SCS recordings performed significantly worse when tested on tonic SCS data. This asymmetry suggests that burst SCS recordings only partially reflect the features present in tonic SCS data, while also capturing additional, potentially more widespread brain dynamics not seen in tonic stimulation. Importantly, this performance gap cannot be solely attributed to inter-patient variability, as recordings from the same patients were included for both stimulation paradigms. One possible explanation is that burst stimulation engages a broader and more complex brain network reflective of stimulation states, which tonic recordings do not capture. This aligns with the hypothesis that tonic stimulation primarily affects the lateral pain pathway, whereas burst stimulation modulates both the medial and lateral pathways [13]. Additionally, the paraesthesias typically induced by tonic stimulation may further contribute to the performance differences observed in the models trained on tonic or burst stimulation recordings.

#### 4.3.3. Recording Sites

Recordings from two different research sites (the Montreal Neurological Institute (MNI) and the Donders Institute) yielded comparable model performance. To further assess generalizability, cross-site evaluation was performed using entirely separate patient groups, ensuring that the model was tested on recordings from individuals not included in the training set. While a slight decrease in accuracy was observed, the GNN models still distinguished between stimulation ON and OFF states. This indicates that the learned representations are robust and capable of generalizing to unseen patients and independent recording environments.

#### 4.3.4. Source Space Analysis

The source-based GNN model showed similar performance to the sensor-based GNN model for the full frequency band. This suggests that the complex transformations involved in source space analysis may not be essential for the GNN to extract relevant differentiating features. However, the source space offers a major advantage for interpretation, since node-level explainability results can be mapped onto anatomically defined brain regions, providing clearer neurophysiological insights. In contrast, sensor-based nodes lack direct anatomical correspondence, limiting interpretability. A comprehensive comparison of different graph structure definitions, including variations in node features and edge construction, could help identify the most effective input representation, whether based on the sensor or source space, from

which the GNN model can extract the most discriminative features.

## 4.4. Model Training

### 4.4.1. Model Hyperparameters

Each input graph configuration resulted in distinct optimal hyperparameter settings, highlighting the sensitivity of the model to the specific characteristics of the input data. In general, models with a greater number of hidden layers tended to yield the highest classification performance during model training. This suggests that higher representational capacity was generally beneficial for learning graph-based patterns in the MEG data. Since a larger number of hidden channels increases the risk of overfitting, particularly when working with relatively small datasets, a trade-off between model complexity and regularization (such as a higher dropout rate) is needed, which is evident in the results of this study. Furthermore, most model configurations performed best with three or four convolutional layers, suggesting that deeper architectures provide better performance for the graph dataset used in this study.

In addition to architectural depth, graph complexity also played a notable role in model performance, particularly the number of retained edges. Edge filtering was applied based on the largest edge weights, effectively pruning weaker or potentially noisier connections. The majority of the best performing model configurations applied edge filtering. The choice of how many edges to retain introduced a meaningful layer of interpretability: retaining more edges increased graph density and potentially allowed the model to capture richer, but also noisier, network information, while more sparse graphs focused the model on the strongest connections. This parameter not only influenced classification accuracy, but also suggests which connections may be most relevant to distinguishing between stimulation states, potentially forming a first step in edge-level explainability. Future research may focus on edge-level explainability approaches to ultimately gain more insights into the specific brain network connections that contribute most significantly to the model's classification performance. Furthermore, when applying stronger edge filtering approaches, either through a higher threshold or a smaller top-K value, a lower dropout rate was found in the model configurations in this study. This suggests that dropout and edge filtering function as complementary regularization techniques with the aim of reducing overfitting.

Nonetheless, the edge-filtering approach relies on the assumption that edges with the largest weights, i.e. the strongest connections, carry the most relevant information for distinguishing between stimulation states. While this may hold true in some cases, it potentially overlooks the importance of more subtle or distributed patterns. The most informative features may not stem from the absolute strengths of individual connections, but rather from broader connectivity patterns or temporal dynamics of connectivity. Thus, focusing solely on the top-weighted edges may obscure the complex brain networks that may characterize effects of SCS. In this context, attention mechanisms within GNNs offer a promising alternative by learning to assign importance to edges or nodes based on their contribution to the classification task, rather than relying solely on predefined metrics such as edge weights. In attention-based GNN architectures, the convolution operation inherently computes updated node embeddings based on the importance of neighbouring nodes by weighing the importance of neighbouring nodes, which is learned during the model learning process [116]. This approach allows the updated embeddings to focus on the most relevant parts of the graph. Even though these attention mechanisms add complexity to the model, these methods might be beneficial to implement in future studies, as this implements an inherent explainability approach to the convolutional layers of the GNN model.

Together, these findings underscore the importance of carefully tuning the parameters for each model configuration, as a balance between model complexity (e.g., number of layers and hidden channels) and

graph complexity (e.g., edge filtering) is crucial for adapting to the specific characteristics of each input configuration.

#### 4.4.2. Model Training

From the training and validation accuracy and loss curves of the sensor- and source-based GNN models, it is evident that both models rapidly learned distinguishing features from the input graphs, with minimal signs of overfitting. In the source-based GNN model for the full band, a temporary instability was observed around iteration 37, likely reflecting a large parameter update or the dropout of a critical subset of features during that specific training iteration. Notably, model performance recovered in the subsequent iterations, suggesting overall training stability.

In contrast, the GNN models trained on the delta, theta, and alpha frequency bands, as well as the combination of theta and alpha bands, exhibited signs of overfitting. This was reflected by a noticeable gap between training and validation accuracy and by a validation loss curve that remained relatively flat while the training loss continued to decrease. A possible explanation for this overfitting is that the input features in these lower frequency bands may lack sufficient discriminative power for the classification task, causing the model to rely on patterns that do not generalize well to unseen data.

Additionally, models trained on the delta band, the combined theta and alpha bands, and recordings from the MNI showed delayed performance improvements compared to the other GNN models. While the delayed performance improvement in the model trained on data from the MNI may be attributed to a lower learning rate setting, the delay observed in models trained on the delta and combined theta–alpha bands is more likely due to the presence of less discriminative features in these frequency ranges.

However, these metrics should be interpreted with caution due to the risk of data leakage, where subepochs from the same patient may be present in both the training and validation sets. This overlap can lead to overly optimistic validation performance. Hence, for assessing the model's generalizability on unseen data, evaluation on an independent external test set is essential.

### 4.5. Node Explainability

To enhance the interpretability of the GNN models, explainability analyses were conducted on sensor-based as well as source-based graphs. Important nodes were identified in tonic and burst recordings from one patient to provide a larger insight into which nodes contributed most to classification decisions.

#### 4.5.1. Important Nodes

Using the sensor- and source-based GNN models across the full frequency band, node-level explainability analyses were conducted on tonic and burst SCS recordings from a single patient. These analyses revealed distinct patterns in the regions contributing to classification performance under each stimulation condition.

For the sensor-based model, tonic stimulation recordings showed that the most informative nodes were roughly located over sensorimotor areas. This finding suggests that neural activity in these regions may be particularly informative for distinguishing stimulation states, with increased pain perception during stimulation OFF potentially driving these changes. This aligns with previous research implicating sensorimotor networks in pain perception and modulation [117]. In the burst stimulation condition, the model exhibited a slightly lower fidelity score, and the most discriminative nodes were more variably distributed, suggesting a less spatially focused network. Despite this variability, the nodes in the identified subgraph roughly implicated sensorimotor regions, indicating a potentially shared brain network involved in burst and tonic stimulation.

The source-based GNN model achieved higher fidelity scores through explainability analysis, indicating that the identified subgraphs played a more decisive role in the model's predictions. In the tonic stimulation recording, two consistently highlighted regions were observed, potentially forming a localized network involving the occipital cortex and motor cortex. Contrarily, for the burst stimulation in the source model, the discriminative subgraph was more spatially distributed, involving the left superior frontal lobe, right anterior insula, right superior parietal cortex, and the anterior cingulate cortex (ACC). The ACC has been widely recognized as a key brain area in chronic pain, often exhibiting elevated baseline activity in patients [118, 119], while the insular cortex is implicated in both sensory and affective aspects of pain processing [120]. The involvement of these regions in the burst stimulation recording suggests that this stimulation paradigm may engage a broader or more diverse set of pain-related brain networks. Taken together with the cross-paradigm performance of the GNN models trained separately on tonic and burst stimulation recordings, these findings support the hypothesis that tonic stimulation primarily influences the lateral pain pathway while burst stimulation modulates both the lateral and medial pathways. The medial pathway is more involved in the affective-emotional dimension of pain, and its engagement during burst stimulation may explain the broader and more distributed network involvement observed in the explainability results. This distinction aligns with previous research and provides additional evidence that burst stimulation engages a more complex and widespread brain network than tonic stimulation [13]. Further research may focus on applying explainability methods to GNN models focussed on either tonic or burst stimulation recordings, with the aim of providing further insights into the active brain networks in both stimulation paradigms.

Interestingly, among the most highlighted nodes, a node that corresponded to the occipital lobe was identified for both recordings, suggesting an involvement of the occipital cortex in pain processing and modulation through SCS. Although this is not a commonly reported brain area related to pain, one study researching the intensity changes of endogenous pain in chronic pain patients found decreased activity in the occipital cortex with higher pain intensities [121]. From this, the hypothesis could be made that occipital cortex activity might reflect pain re-emerging after the stimulation has been deactivated. This hypothesis should be researched in future studies.

Notably, the explainability results from the sensor-based GNN model identified key nodes predominantly located in the left hemisphere for both tonic and burst recordings. In contrast, the source-based GNN model highlighted brain areas primarily in the right hemisphere during tonic stimulation. Given that the patient *PT06* experienced chronic pain on both the left and right sides of the body, this hemispheric discrepancy is unlikely to be solely driven by the lateralization of the pain. Rather, it may result from the fact that the sensor- and source-based GNN models were trained separately and could have learned to rely on different discriminative features. Each model may capture distinct aspects of the underlying neural activity and identify different functional brain networks that contribute to classification performance.

#### 4.5.2. Interpretation

Taken together, the results of the node explainability analysis suggest that the GNN models primarily rely on regions involved in pain processing. These nodes appear to contribute the most discriminative features for the classification task, potentially highlighting the modulation of key pain-processing regions by SCS. Consequently, excluding these nodes and their node features leads to a substantial reduction in the model's prediction confidence, highlighting their importance in the decision-making process.

However, it is important to carefully interpret the node-level explainability results from this study. Since the analysis was conducted on a single patient, only a limited number of graphs were available for evaluation, possibly containing patient-specific features, which limits the generalizability of the explainability results.

Moreover, each graph yielded slightly different results, both in terms of which nodes were identified as most important and the associated fidelity scores. This variability underscores the need for careful consideration when drawing conclusions from individual graph interpretations. As such, the identified important nodes may primarily reflect individual neural patterns rather than general brain networks that are activated during chronic pain and SCS. In contrast, applying explainability techniques across a larger group of patients in future studies could help identify more consistent and generalizable patterns of node importance. This would allow for the distinction between individual variability and shared neural mechanisms underlying SCS effects, providing more robust insights into the brain regions most consistently involved in pain modulation.

Adding to this, the explainability approach that was implemented using SubgraphX was initially developed for GNN models with node classification architectures [62]. Therefore, adaptations to the initially proposed explainability method were required to make it applicable to GNN models for graph classification. Nevertheless, a method specifically designed for explainability in graph classification tasks may be better suited to fully leverage the underlying graph structure, potentially including both node features and edge weights, and thereby more accurately identify the most relevant attributes contributing to the model's decisions.

Through scaling this approach to a larger cohort could provide valuable insights into the supraspinal mechanisms of SCS and its role in modulating pain processing in the brain. Beyond interpretability, these tools also help identify the most informative nodes and graph components, guiding model optimization strategies aimed at reducing computational load while preserving or potentially even enhancing classification performance.

## 4.6. General Strengths and Limitations

This study introduces a novel approach to investigating chronic pain and SCS by applying GNNs to resting-state MEG data. To the best of our knowledge, this is the first study to employ GNNs for classifying stimulation states and identifying pain-related brain connectivity patterns in this context. A key advantage of GNNs is their ability to move beyond manually defined features, which traditional machine learning models heavily rely on. While GNNs still require an initial definition of graph topology and node or edge attributes, they can autonomously learn higher-order representations through the convolutional layers. This internal feature learning enables GNNs to capture complex relational and spatial patterns, making them particularly suited for analysing high-dimensional, interconnected data such as brain connectivity networks. Consequently, GNNs offer a powerful and flexible approach for extracting meaningful patterns from MEG data in the context of chronic pain and SCS.

### 4.6.1. Analysis Framework

A major strength of this work lies in the flexibility and modularity of the analysis framework. By incorporating both sensor-level and source-level data, and designing the graph dataset and GNN model in a highly adaptable way, the framework allows for straightforward modification of input types and parameter settings. This modular design facilitates the exploration of various configurations, making it well-suited for extending the analysis to different frequency bands and other graph attributes. This enables a thorough evaluation of model performance across different graph-based representations of the MEG signals. The model's performance was evaluated across a range of input configurations, including multiple frequency bands, stimulation paradigms, and even differences in recording institutes. The external validation through cross-institute testing further enhances the model's generalizability by assessing performance on unseen data from a different patient cohort.

### 4.6.2. Design Choices

However, despite these strengths, certain limitations must be considered. Firstly, while a wide range of input configurations was tested, some design choices were made early in the study and not fully explored. This limited the ability to assess the impact of various graph attributes (e.g., node features or edge definitions) on model performance, which could have provided more granular insights into how specific graph properties influence classification outcomes. Additionally, the temporal resolution used in the current analysis was not extensively optimized, which may be relevant for capturing the dynamic fluctuations in functional connectivity that are known to characterize chronic pain states [122]. While the use of 30-second sliding windows with 25-second overlap provides a balance between temporal sensitivity and stability of the graph attributes, future work could comprehensively explore the impact of varying window lengths on different graph attributes to better capture transient but potentially informative neural dynamics.

Another limitation is that the GNN model architecture was developed and optimized for the full frequency band, and the same architecture was applied to all subsequent models. For input configurations involving narrower frequency bands, the current model architecture may not be ideally suited. Re-optimizing the GNN models for each frequency band could improve classification performance and enable more meaningful comparisons across bands.

### 4.6.3. Dataset

Furthermore, the relatively small dataset, which is constrained by the limited number of patient recordings, reduces the generalizability of the findings. The high inter-patient variability, particularly in chronic pain populations, also complicates drawing standardized conclusions. The current dataset predominantly includes patients with Persistent Spinal Pain Syndrome type 2 (PSPS-2), which limits the exploration of connectivity patterns across other chronic pain conditions. Differences in the anatomical location of pain among patients result in stimulation being applied at different spinal levels, reflecting variability in how SCS is implemented across individuals. Additionally, the variation in pain scores in response to the different SCS paradigms highlights the differences in the effectiveness of SCS treatment between patients. More personalized approaches may be considered to account for these inter-individual differences in pain processing and treatment response. Through incorporating patient-specific information, such as pain condition, pain intensity, pain reduction for each stimulation paradigm, as auxiliary features in the model, the learning process could better capture individual variability in pain mechanisms and responses to SCS. Such personalized models may enhance the identification of individualized patterns that are characteristic of SCS efficacy across different pain conditions, potentially leading to more targeted and effective treatment strategies.

Additionally, this study did not consider the anatomical location of SCS, which could play a significant role in shaping both the neural response to stimulation and the observed connectivity patterns. Differences in the anatomical location of pain among patients result in stimulation being applied at different spinal levels, reflecting variability in how SCS is implemented. As a result, individual differences in lead placement and targeted dermatomes could contribute to variability in connectivity-based network alterations. Future studies should incorporate stimulation site information to better account for these inter-individual differences. One approach could involve stratifying patients based on the anatomical location of SCS and analysing each group separately, which may help identify location-specific neural responses and connectivity patterns. Alternatively, stimulation site could be included as an additional auxiliary feature, allowing the GNN model to adjust for potential variations. Such approaches may improve the interpretability of results and support more personalized insights into how stimulation modulates

brain networks. However, to enable meaningful subgroup analyses across different stimulation locations, a sufficiently large dataset of MEG recordings, with approximately 20 patients for each stimulation site, would be necessary.

Building on this, it is important to note that the proposed GNN model architecture does not incorporate patient-stratified data splits, as it was not explicitly ensured that graphs from the same patient were excluded from both the training and test sets. This introduces a potential risk of data leakage, which may have significantly inflated the reported performance metrics. This issue stems from the fact that multiple graphs were generated per patient, capturing different stimulation states or brain activity patterns. If graphs from the same patient are present in both the training and test sets, the model may inadvertently learn patient-specific features rather than generalizable patterns related to the classification task. As a result, the model's performance on the test set may not reflect its true ability to generalize to unseen patients, but rather its familiarity with individual-specific characteristics already encountered during training.

To address the potential impact of data leakage and better assess the model's generalizability, I included evaluation of the cross-paradigm and cross-institute performance metrics. In particular, cross-institute performance was valuable for testing the model on a completely independent set of patients. This setup offers a more robust assessment of model performance compared to testing on a single external patient file or using a test split without stratification on patient level, as it eliminates overlap between training and testing data and better reflects real-world variability. Consequently, these outcomes provide stronger evidence for the model's ability to generalize across different patient populations and recording environments. Nonetheless, further research should implement stratification in the data split to ensure that graphs originating from the same patient cannot be present in both training and test sets. However, ensuring that graphs from the same patient are not present in both the training and test sets becomes more complex in this study, as MEG recordings from a single patient were included for different stimulation paradigms (tonic and burst stimulation). These recordings, although representing different stimulation conditions, are still derived from the same individual, meaning they could share patient-specific neural characteristics. To fully account for this, future research should consider evaluating tonic and burst recordings using separate GNN models for each stimulation paradigm. This approach would allow for paradigm-specific differentiation, reducing the risk of data leakage by ensuring that the model does not inadvertently learn shared patient-specific characteristics from both stimulation paradigms.

#### **4.6.4. Computational Demands**

From a technical standpoint, the computational demands associated with processing and training the GNN model are substantial, particularly due to the number of nodes and the range of frequencies, both of which significantly increase the computational load. However, once the initial graph construction is completed, including the resource-intensive computation of graph attributes, the resulting graphs provide an efficient and compact way to store MEG-based representations, enabling more streamlined model training and evaluation. Moreover, the explainability module was especially resource-intensive, requiring evaluation of each graph individually. As a result, this analysis was limited to data from a single patient, restricting the generalizability of the interpretability findings. Future work should focus on implementing a more efficient explainability method for analysis across a larger and more diverse set of patients to enhance the generalizability and reliability of interpretability findings, potentially identifying consistent important nodes for the model's decision-making process. These outcomes could then be leveraged to further reduce the computational load by focusing solely on the most relevant and distinctive components of the input graphs.

#### 4.6.5. Carry-over Effects

A key physiological consideration is the carry-over effect associated with SCS. When stimulation is turned off, the return of pain perception does not occur immediately. This delay has been reported to last over two hours in some cases [65], far longer than the one-minute intervals used in this study. Consequently, segments labelled with stimulation OFF may still reflect residual effects of prior stimulation, complicating classification and interpretation. The carry-over period is thought to be longer for burst SCS than for tonic stimulation, which further complicates the analysis. Conversely, a reverse carry-over effect exists, where pain relief after reactivating the stimulator takes time to re-establish. In addition to these delayed effects on pain perception, it should be considered that SCS may exert direct neuromodulatory effects on brain activity, independent of the patient's experience of pain. This implies that ongoing stimulation could induce immediate changes in brain connectivity or network dynamics, even before alterations in perceived pain are reported. As a result, the GNN models may focus on differentiation of both pain-related brain activity and direct stimulation-related effects, complicating the disentanglement of pain relief mechanisms from the effects of SCS.

### 4.7. Recommendations

Since the outcomes of this study cannot completely differentiate between the effects of the underlying chronic pain condition and the SCS effects, future research should focus on applying GNN models for classification of active brain networks in chronic pain versus pain-free controls. This would help identify network-level connectivity alterations distinguishing chronic pain from normal brain function. Additionally, differentiating between the presence and absence of pain in chronic pain patients undergoing SCS could provide valuable insights into the effects of SCS on brain networks. To achieve this, patients could be grouped into responders and non-responders based on their reported pain scores with SCS. A GNN model could then be trained to classify brain networks into responders and non-responders, helping to uncover the neural differences between these groups. This approach could provide further insights into the mechanisms of chronic pain and the functional connectivity patterns that are indicative of the effectiveness of SCS. Ultimately, this could help personalize and optimize pain management strategies based on individual brain network characteristics. In the future, this could possibly assist in identifying patients that will have the most effect of SCS, prior to implantation.

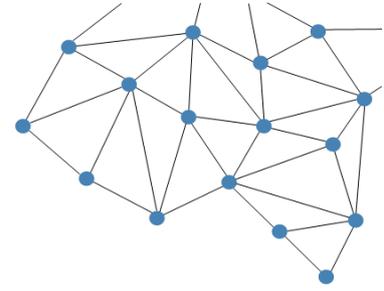
Additionally, future studies should aim to systematically explore and evaluate a broader range of input configurations, including various graph attributes (e.g., node features and edge definitions), as well as different graph sizes through varying in the numbers of nodes. A more comprehensive assessment of these attributes and graph sizes will provide valuable insights into how different graph properties influence model performance and may lead to better-informed decisions about an optimal model design that allows interpretation of the outcomes.

For a deeper understanding of the distinct effects of tonic versus burst stimulation, it would be valuable to analyse separate GNN models for each stimulation paradigm. Additionally, conducting explainability analyses on these models could help uncover the underlying mechanisms driving the differential effects of these stimulation paradigms. This could potentially provide insights that help determine which stimulation paradigm is most effective based on individual patient features, thereby assisting in the optimization of treatment.

Furthermore, implementation of more extensive explainability analysis, involving the entire graph structure, is also a topic of large interest for future research. A more extensive explainability analysis might identify the most important node features, as well as the most important edges that are largely informative for

---

the model's decision-making process. This analysis could indicate the exact brain networks that are discriminative for classification between stimulation states. Clinically relevant mechanisms of chronic pain and the treatment targets could be grasped by utilizing these extensive explainability analyses for GNN models aiming to classify chronic pain and pain-free controls, as well as GNN models differentiating the presence and absence of pain during SCS.



## 5 | Conclusion

This thesis introduces a novel application of GNNs to classify active brain networks in chronic pain patients treated with SCS using resting-state MEG data. The study is the first to apply GNNs for classifying stimulation states associated with brain network patterns in this context, it marks a significant advancement in the network-level analysis of chronic pain and neuromodulation. By leveraging the flexibility of the GNN model architecture, various graph representations of brain activity based on MEG data were constructed, using both sensor space and source space analysis. GNN models trained on these representations achieved high classification performance in distinguishing between SCS ON and OFF states.

Model performance analysis across different frequency bands revealed that the beta and gamma bands, as well as the full frequency band (1 - 100 Hz), carried the most discriminative information, reaching test accuracies of 0.98 for the beta and gamma bands and 0.99 for the full band. Differences between tonic and burst SCS paradigms were assessed by training and testing separate GNN models. Both paradigms yielded comparable classification results with test accuracies of 0.98 and 0.97, respectively, however, cross-paradigm evaluation revealed asymmetry: the GNN model trained on burst SCS data underperformed when tested on tonic SCS data (cross-paradigm accuracy of 0.69), whereas the GNN model trained on tonic SCS data generalized better to burst SCS recordings (cross-paradigm accuracy of 0.88). This may reflect the broader network modulation associated with burst SCS. Furthermore, generalizability of the GNN model was supported by training and testing GNN models on data from different recording institutes, showing consistent performance across datasets with cross-site accuracies of 0.81 and 0.87.

Node-level explainability analysis highlighted discriminative nodes in known pain-related brain regions, supporting the notion that SCS modulates these networks. However, interpretation is limited due to inconsistencies in node explainability outcomes across sensor- and source-space GNN models, as well as variability between individual graph inputs. Moreover, given that the analysis was performed on a small number of patients, it remains difficult to disentangle effects of the underlying chronic pain condition from the effects induced by turning stimulation ON and OFF. Additionally, variability in patient-specific carry-over effects of stimulation adds further complexity to these findings.

The results of this thesis represent a significant step in applying GNNs to chronic pain and SCS research, highlighting their potential to unravel complex brain network dynamics and providing an analysis framework for further refinement and exploration in this research field.

# References

- [1] R.-D. Treede *et al.*, “A classification of chronic pain for ICD-11,” eng, *Pain*, vol. 156, no. 6, pp. 1003–1007, Jun. 2015, ISSN: 1872-6623. DOI: 10.1097/j.pain.000000000000160.
- [2] R.-D. Treede *et al.*, “Chronic pain as a symptom or a disease: The IASP Classification of Chronic Pain for the International Classification of Diseases (ICD-11),” eng, *Pain*, vol. 160, no. 1, pp. 19–27, Jan. 2019, ISSN: 1872-6623. DOI: 10.1097/j.pain.0000000000001384.
- [3] J. Barchini *et al.*, “Spinal segmental and supraspinal mechanisms underlying the pain-relieving effects of spinal cord stimulation: An experimental study in a rat model of neuropathy,” *Neuroscience*, vol. 215, pp. 196–208, Jul. 2012, ISSN: 0306-4522. DOI: 10.1016/j.neuroscience.2012.04.057.
- [4] C. El-Khoury, N. Hawwa, M. Baliki, S. F. Atweh, S. J. Jabbur, and N. E. Saadé, “Attenuation of neuropathic pain by segmental and supraspinal activation of the dorsal column system in awake rats,” eng, *Neuroscience*, vol. 112, no. 3, pp. 541–553, 2002, ISSN: 0306-4522. DOI: 10.1016/s0306-4522(02)00111-2.
- [5] K. V. Slavin, R. B. North, T. R. Deer, P. Staats, K. Davis, and R. Diaz, “Tonic and burst spinal cord stimulation waveforms for the treatment of chronic, intractable pain: Study protocol for a randomized controlled trial,” *Trials*, vol. 17, no. 1, p. 569, Dec. 2016, ISSN: 1745-6215. DOI: 10.1186/s13063-016-1706-5.
- [6] O. Viswanath, I. Urits, E. Bouley, J. M. Peck, W. Thompson, and A. D. Kaye, “Evolving Spinal Cord Stimulation Technologies and Clinical Implications in Chronic Pain Management,” en, *Current Pain and Headache Reports*, vol. 23, no. 6, p. 39, May 2019, ISSN: 1534-3081. DOI: 10.1007/s11916-019-0778-9.
- [7] D. De Ridder, S. Vanneste, M. Plazier, E. Van Der Loo, and T. Menovsky, “Burst spinal cord stimulation: Toward paresthesia-free pain suppression,” *Neurosurgery*, vol. 66, no. 5, pp. 986–990, 2010. DOI: 10.1227/01.NEU.0000368153.44883.B3.
- [8] H. Swadlow and A. Gusev, “The impact of ‘bursting’ thalamic impulses at a neocortical synapse,” *Nature Neuroscience*, vol. 4, no. 4, pp. 402–408, 2001. DOI: 10.1038/86054.
- [9] D. De Ridder *et al.*, “A 2-center comparative study on tonic versus burst spinal cord stimulation: Amount of responders and amount of pain suppression,” eng, *The Clinical Journal of Pain*, vol. 31, no. 5, pp. 433–437, May 2015, ISSN: 1536-5409. DOI: 10.1097/AJP.000000000000129.
- [10] J.-B. Peeters and C. Raftopoulos, “Tonic, Burst, High-Density, and 10-kHz High-Frequency Spinal Cord Stimulation: Efficiency and Patients’ Preferences in a Failed Back Surgery Syndrome Predominant Population. Review of Literature,” eng, *World Neurosurgery*, vol. 144, e331–e340, Dec. 2020, ISSN: 1878-8769. DOI: 10.1016/j.wneu.2020.08.128.
- [11] C. C. de Vos, M. J. Bom, S. Vanneste, M. W. P. M. Lenders, and D. de Ridder, “Burst spinal cord stimulation evaluated in patients with failed back surgery syndrome and painful diabetic neuropathy,” eng, *Neuromodulation: Journal of the International Neuromodulation Society*, vol. 17, no. 2, pp. 152–159, Feb. 2014, ISSN: 1525-1403. DOI: 10.1111/ner.12116.
- [12] D. De Ridder, D. Adhia, and S. Vanneste, “The anatomy of pain and suffering in the brain and its clinical implications,” eng, *Neuroscience and Biobehavioral Reviews*, vol. 130, pp. 125–146, Nov. 2021, ISSN: 1873-7528. DOI: 10.1016/j.neubiorev.2021.08.013.

- [13] D. De Ridder and S. Vanneste, "Burst and Tonic Spinal Cord Stimulation: Different and Common Brain Mechanisms," eng, *Neuromodulation: Journal of the International Neuromodulation Society*, vol. 19, no. 1, pp. 47–59, Jan. 2016, ISSN: 1525-1403. DOI: 10.1111/ner.12368.
- [14] D. De Ridder, M. Plazier, N. Kamerling, T. Menovsky, and S. Vanneste, "Burst spinal cord stimulation for limb and back pain," eng, *World Neurosurgery*, vol. 80, no. 5, 642–649.e1, Nov. 2013, ISSN: 1878-8769. DOI: 10.1016/j.wneu.2013.01.040.
- [15] S.-L. Li, J. Li, H.-C. Xu, Y.-C. Liu, T.-T. Yang, and H. Yuan, "Progress in the efficacy and mechanism of spinal cord stimulation in neuropathological pain," en, *Ibrain*, vol. 8, no. 1, pp. 23–36, 2022, \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ibra.12020>, ISSN: 2769-2795. DOI: 10.1002/ibra.12020.
- [16] K. Kumar, G. Hunter, and D. Demeria, "Spinal cord stimulation in treatment of chronic benign pain: Challenges in treatment planning and present status, a 22-year experience," eng, *Neurosurgery*, vol. 58, no. 3, 481–496, discussion 481–496, Mar. 2006, ISSN: 1524-4040. DOI: 10.1227/01.NEU.0000192162.99567.96.
- [17] E. A. Joosten and G. Franken, "Spinal cord stimulation in chronic neuropathic pain: Mechanisms of action, new locations, new paradigms," *Pain*, vol. 161, no. 1, S104–S113, Sep. 2020, ISSN: 0304-3959. DOI: 10.1097/j.pain.0000000000001854.
- [18] O. Sporns, "Graph theory methods: Applications in brain networks," en, *Dialogues in Clinical Neuroscience*, vol. 20, no. 2, p. 111, Jun. 2018, Publisher: Taylor & Francis. DOI: 10.31887/DCNS.2018.20.2/osporns.
- [19] J. Jin *et al.*, "A Novel Classification Framework Using the Graph Representations of Electroencephalogram for Motor Imagery Based Brain-Computer Interface," eng, *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 30, pp. 20–29, 2022, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2021.3139095.
- [20] C. J. Stam *et al.*, "Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease," *Brain*, vol. 132, no. 1, pp. 213–224, Jan. 2009, ISSN: 0006-8950. DOI: 10.1093/brain/awn262.
- [21] C. Stam, B. Jones, G. Nolte, M. Breakspear, and P. Scheltens, "Small-world networks and functional connectivity in Alzheimer's disease," *Cerebral Cortex*, vol. 17, no. 1, pp. 92–99, 2007. DOI: 10.1093/cercor/bhj127.
- [22] D. S. Bassett and O. Sporns, "Network neuroscience," eng, *Nature Neuroscience*, vol. 20, no. 3, pp. 353–364, Feb. 2017, ISSN: 1546-1726. DOI: 10.1038/nn.4502.
- [23] S. Vanneste and D. De Ridder, "Chronic pain as a brain imbalance between pain input and pain suppression," *Brain Communications*, vol. 3, no. 1, fcab014, Feb. 2021, ISSN: 2632-1297. DOI: 10.1093/braincomms/fcab014.
- [24] K. J. Friston, "Functional and effective connectivity in neuroimaging: A synthesis," en, *Human Brain Mapping*, vol. 2, no. 1-2, pp. 56–78, 1994, ISSN: 1097-0193. DOI: 10.1002/hbm.460020107.
- [25] O. Sporns, G. Tononi, and R. Kötter, "The Human Connectome: A Structural Description of the Human Brain," en, *PLoS Computational Biology*, vol. 1, no. 4, e42, 2005, ISSN: 1553-734X, 1553-7358. DOI: 10.1371/journal.pcbi.0010042.
- [26] F. De Vico Fallani, J. Richiardi, M. Chavez, and S. Achard, "Graph analysis of functional brain networks: Practical issues in translational neuroscience," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1653, p. 20130521, Oct. 2014. DOI: 10.1098/rstb.2013.0521.
- [27] J. D. Bonita *et al.*, "Time domain measures of inter-channel EEG correlations: A comparison of linear, nonparametric and nonlinear measures," en, *Cognitive Neurodynamics*, vol. 8, no. 1, pp. 1–15, Feb. 2014, ISSN: 1871-4099. DOI: 10.1007/s11571-013-9267-8.

- [28] C. J. Stam, "Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field," *Clinical Neurophysiology*, vol. 116, no. 10, pp. 2266–2301, Oct. 2005, ISSN: 1388-2457. DOI: 10.1016/j.clinph.2005.06.011.
- [29] E. Pereda, R. Q. Quiroga, and J. Bhattacharya, "Nonlinear multivariate analysis of neurophysiological signals," *Progress in Neurobiology*, vol. 77, no. 1, pp. 1–37, Sep. 2005, ISSN: 0301-0082. DOI: 10.1016/j.pneurobio.2005.10.003.
- [30] G. Chiarion, L. Sparacino, Y. Antonacci, L. Faes, and L. Mesin, "Connectivity Analysis in EEG Data: A Tutorial Review of the State of the Art and Emerging Trends," en, *Bioengineering*, vol. 10, no. 3, p. 372, Mar. 2023, ISSN: 2306-5354. DOI: 10.3390/bioengineering10030372.
- [31] T. C. Zhang, J. J. Janik, and W. M. Grill, "Mechanisms and models of spinal cord stimulation for the treatment of neuropathic pain," eng, *Brain Research*, vol. 1569, pp. 19–31, Jun. 2014, ISSN: 1872-6240. DOI: 10.1016/j.brainres.2014.04.039.
- [32] M. Moens *et al.*, "Spinal cord stimulation modulates cerebral function: An fMRI study," en, *Neuroradiology*, vol. 54, no. 12, pp. 1399–1407, Dec. 2012, ISSN: 1432-1920. DOI: 10.1007/s00234-012-1087-8.
- [33] Z. A. Gaál, R. Boha, C. J. Stam, and M. Molnár, "Age-dependent features of EEG-reactivity—Spectral, complexity, and network characteristics," *Neuroscience Letters*, vol. 479, no. 1, pp. 79–84, Jul. 2010, ISSN: 0304-3940. DOI: 10.1016/j.neulet.2010.05.037.
- [34] F. Vecchio, F. Miraglia, P. Bramanti, and P. M. Rossini, "Human Brain Networks in Physiological Aging: A Graph Theoretical Analysis of Cortical Connectivity from EEG Data," en, *Journal of Alzheimer's Disease*, vol. 41, no. 4, pp. 1239–1249, Jan. 2014, ISSN: 1387-2877. DOI: 10.3233/JAD-140090.
- [35] C. J. Stam and J. C. Reijneveld, "Graph theoretical analysis of complex networks in the brain," *Nonlinear Biomedical Physics*, vol. 1, no. 1, p. 3, Jul. 2007, ISSN: 1753-4631. DOI: 10.1186/1753-4631-1-3.
- [36] A. Griffa, P. S. Baumann, J.-P. Thiran, and P. Hagmann, "Structural connectomics in brain diseases," *NeuroImage*, Mapping the Connectome, vol. 80, pp. 515–526, Oct. 2013, ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.04.056.
- [37] Q. K. Telesford, S. L. Simpson, J. H. Burdette, S. Hayasaka, and P. J. Laurienti, "The Brain as a Complex System: Using Network Science as a Tool for Understanding the Brain," *Brain Connectivity*, vol. 1, no. 4, pp. 295–308, Oct. 2011, ISSN: 2158-0014. DOI: 10.1089/brain.2011.0055.
- [38] F. V. Farahani, W. Karwowski, and N. R. Lighthall, "Application of Graph Theory for Identifying Connectivity Patterns in Human Brain Networks: A Systematic Review," English, *Frontiers in Neuroscience*, vol. 13, Jun. 2019, ISSN: 1662-453X. DOI: 10.3389/fnins.2019.00585.
- [39] J. Barroso, P. Branco, and A. V. Apkarian, "Brain mechanisms of chronic pain: Critical role of translational approach," *Translational research : the journal of laboratory and clinical medicine*, vol. 238, pp. 76–89, Dec. 2021, ISSN: 1931-5244. DOI: 10.1016/j.trsl.2021.06.004.
- [40] D. Lenoir, B. Cagnie, H. Verhelst, and R. D. Pauw, "Graph Measure Based Connectivity in Chronic Pain Patients: A Systematic Review," en, *Pain Physician*, 2021.
- [41] C. L. Alves *et al.*, "Analysis of functional connectivity using machine learning and deep learning in different data modalities from individuals with schizophrenia," eng, *Journal of Neural Engineering*, vol. 20, no. 5, Sep. 2023, ISSN: 1741-2552. DOI: 10.1088/1741-2552/acf734.
- [42] W. Liu, P. Fang, F. Guo, Y. Qiao, Y. Zhu, and H. Wang, "Graph-Theory-Based Degree Centrality Combined with Machine Learning Algorithms Can Predict Response to Treatment with Antipsychotic Medications in Patients with First-Episode Schizophrenia," *Disease Markers*, vol. 2022, p. 1 853 002, Oct. 2022, ISSN: 0278-0240. DOI: 10.1155/2022/1853002.
- [43] Pooja, S. K. Pahuja, and K. Veer, "Recent Approaches on Classification and Feature Extraction of EEG Signal: A Review," en, *Robotica*, vol. 40, no. 1, pp. 77–101, Jan. 2022, ISSN: 0263-5747, 1469-8668. DOI: 10.1017/S0263574721000382.

- [44] J. Shen, X. Zhang, G. Wang, Z. Ding, and B. Hu, "An Improved Empirical Mode Decomposition of Electroencephalogram Signals for Depression Detection," *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 262–271, Jan. 2022, ISSN: 1949-3045. DOI: 10.1109/TAFFC.2019.2934412.
- [45] J. Shen *et al.*, "Exploring the Intrinsic Features of EEG Signals via Empirical Mode Decomposition for Depression Recognition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 356–365, 2023, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2022.3221962.
- [46] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces," *eng, Journal of Neural Engineering*, vol. 15, no. 5, p. 056 013, Oct. 2018, ISSN: 1741-2552. DOI: 10.1088/1741-2552/aace8c.
- [47] R. G. Hefron, B. J. Borghetti, J. C. Christensen, and C. M. S. Kabban, "Deep long short-term memory structures model temporal dependencies improving cognitive workload estimation," *en, Pattern Recognition Letters*, vol. 94, pp. 96–104, Jul. 2017, ISSN: 01678655. DOI: 10.1016/j.patrec.2017.05.020.
- [48] M. A. Myszczyńska *et al.*, "Applications of machine learning to diagnosis and treatment of neurodegenerative diseases," *eng, Nature Reviews. Neurology*, vol. 16, no. 8, pp. 440–456, Aug. 2020, ISSN: 1759-4766. DOI: 10.1038/s41582-020-0377-8.
- [49] A. Abrol *et al.*, "Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning," *en, Nature Communications*, vol. 12, no. 1, p. 353, Jan. 2021, ISSN: 2041-1723. DOI: 10.1038/s41467-020-20655-6.
- [50] M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017. DOI: 10.1109/MSP.2017.2693418.
- [51] P. Cui, X. Wang, J. Pei, and W. Zhu, "A Survey on Network Embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, May 2019, ISSN: 1558-2191. DOI: 10.1109/TKDE.2018.2849727.
- [52] D. Zhang, C.-Y. Chow, A. Liu, X. Zhang, Q. Ding, and Q. Li, "Efficient evaluation of shortest travel-time path queries through spatial mashups," *en, Geoinformatica*, vol. 22, no. 1, pp. 3–28, Jan. 2018, ISSN: 1573-7624. DOI: 10.1007/s10707-016-0288-4.
- [53] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," vol. 2, Jan. 2005, 729–734 vol. 2, ISBN: 978-0-7803-9048-5. DOI: 10.1109/IJCNN.2005.1555942.
- [54] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *en, IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009, ISSN: 1045-9227, 1941-0093. DOI: 10.1109/TNN.2008.2005605.
- [55] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020, ISSN: 2666-6510. DOI: 10.1016/j.aiopen.2021.01.001.
- [56] A. T. Adebisi, H.-W. Lee, and K. C. Veluvolu, "EEG-Based Brain Functional Network Analysis for Differential Identification of Dementia-Related Disorders and Their Onset," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 1198–1209, 2024, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2024.3374651.
- [57] D. Klepl, M. Wu, and F. He, *Graph Neural Network-based EEG Classification: A Survey*, *en*, Dec. 2023.
- [58] D. Klepl, F. He, M. Wu, D. J. Blackburn, and P. Sarrigiannis, "EEG-Based Graph Neural Network Classification of Alzheimer's Disease: An Empirical Evaluation of Functional Connectivity Methods," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2651–2660, 2022, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2022.3204913.
- [59] X. Shan, J. Cao, S. Huo, L. Chen, P. G. Sarrigiannis, and Y. Zhao, "Spatial-temporal graph convolutional network for Alzheimer classification based on brain functional connectivity imaging of electroen-

- cephalogram,” en, *Human Brain Mapping*, vol. 43, no. 17, pp. 5194–5209, 2022, ISSN: 1097-0193. DOI: 10.1002/hbm.25994.
- [60] C. F. Witstok, “Connectivity Analysis in Chronic Pain using Graph Neural Networks: A review,” Erasmus Medical Center, Rotterdam, Tech. Rep., 2024.
- [61] H. Yuan, J. Tang, X. Hu, and S. Ji, “XGNN: Towards Model-Level Explanations of Graph Neural Networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’20, New York, NY, USA: Association for Computing Machinery, Aug. 2020, pp. 430–438, ISBN: 978-1-4503-7998-4. DOI: 10.1145/3394486.3403085.
- [62] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, “On Explainability of Graph Neural Networks via Subgraph Explorations,” en, in *Proceedings of the 38th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jul. 2021, pp. 12 241–12 252.
- [63] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “GNNExplainer: Generating Explanations for Graph Neural Networks,” in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.
- [64] D. Luo *et al.*, *Parameterized Explainer for Graph Neural Network*, arXiv:2011.04573 [cs], Nov. 2020. DOI: 10.48550/arXiv.2011.04573.
- [65] K. Meier *et al.*, “Examining the Duration of Carryover Effect in Patients With Chronic Pain Treated With Spinal Cord Stimulation (EChO Study): An Open, Interventional, Investigator-Initiated, International Multicenter Study,” *Neuromodulation: Technology at the Neural Interface*, vol. 27, no. 5, pp. 887–898, Jul. 2024, ISSN: 1094-7159. DOI: 10.1016/j.neurom.2024.01.002.
- [66] *CTF MEG, Canada*, en. [Online]. Available: <https://www.ctf.com> (visited on 03/28/2025).
- [67] F. Tadel, S. Baillet, J. C. Mosher, D. Pantazis, and R. M. Leahy, “Brainstorm: A user-friendly application for MEG/EEG analysis,” eng, *Computational Intelligence and Neuroscience*, vol. 2011, p. 879 716, 2011, ISSN: 1687-5273. DOI: 10.1155/2011/879716.
- [68] *MNE – MNE 1.9.0 documentation*. [Online]. Available: <https://mne.tools/stable/index.html> (visited on 01/22/2025).
- [69] *PyG Documentation – pytorch\_geometric documentation*. [Online]. Available: <https://pytorch-geometric.readthedocs.io/en/latest/> (visited on 03/28/2025).
- [70] *Scale Machine Learning & AI Computing | Ray by Anyscale*, en-US. [Online]. Available: <https://ray.io> (visited on 03/28/2025).
- [71] *Tutorials/BadChannels - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/BadChannels> (visited on 03/28/2025).
- [72] A. de Cheveigné and I. Nelken, “Filters: When, Why, and How (Not) to Use Them,” eng, *Neuron*, vol. 102, no. 2, pp. 280–293, Apr. 2019, ISSN: 1097-4199. DOI: 10.1016/j.neuron.2019.02.039.
- [73] *Tutorials/SourceEstimation - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/SourceEstimation> (visited on 03/28/2025).
- [74] *Tutorials/Workflows - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/Workflows> (visited on 03/28/2025).
- [75] *Tutorials/HeadModel - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/HeadModel> (visited on 03/28/2025).
- [76] C. Destrieux, B. Fischl, A. Dale, and E. Halgren, “Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature,” *NeuroImage*, vol. 53, no. 1, pp. 1–15, Oct. 2010, ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2010.06.010.
- [77] I. Tracey, “Imaging pain,” *British Journal of Anaesthesia*, vol. 101, no. 1, pp. 32–39, Jul. 2008, ISSN: 0007-0912. DOI: 10.1093/bja/aen102.
- [78] Q. Chang *et al.*, “Classification of First-Episode Schizophrenia, Chronic Schizophrenia and Healthy Control Based on Brain Network of Mismatch Negativity by Graph Neural Network,” *IEEE Transactions*

- on *Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1784–1794, 2021, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2021.3105669.
- [79] W. Hu, G. Jiang, J. Han, X. Li, and P. Xie, “Regional-Asymmetric Adaptive Graph Convolutional Neural Network for Diagnosis of Autism in Children With Resting-State EEG,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 200–211, 2024, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2023.3347134.
- [80] J. Cao, L. Yang, P. G. Sarrigiannis, D. Blackburn, and Y. Zhao, “Dementia classification using a graph neural network on imaging of effective brain connectivity,” *Computers in Biology and Medicine*, vol. 168, p. 107 701, Jan. 2024, ISSN: 0010-4825. DOI: 10.1016/j.combiomed.2023.107701.
- [81] P. E. Vlisides *et al.*, “Dynamic Cortical Connectivity during General Anesthesia in Surgical Patients,” *eng, Anesthesiology*, vol. 130, no. 6, pp. 885–897, Jun. 2019, ISSN: 1528-1175. DOI: 10.1097/ALN.0000000000002677.
- [82] D. M. Groppe *et al.*, “Dominant frequencies of resting human brain activity as measured by the electrocorticogram,” *en, NeuroImage*, vol. 79, p. 223, Apr. 2013. DOI: 10.1016/j.neuroimage.2013.04.044.
- [83] H. Zhou *et al.*, “Effect of Implantable Electrical Nerve Stimulation on Cortical Dynamics in Patients With Herpes Zoster–Related Pain: A Prospective Pilot Study,” *English, Frontiers in Bioengineering and Biotechnology*, vol. 10, May 2022, Publisher: Frontiers, ISSN: 2296-4185. DOI: 10.3389/fbioe.2022.862353.
- [84] L. Chen *et al.*, “Correlation between spinal cord stimulation analgesia and cortical dynamics in pain management,” *Annals of Clinical and Translational Neurology*, vol. 11, no. 1, pp. 57–66, Oct. 2023, ISSN: 2328-9503. DOI: 10.1002/acn3.51932.
- [85] K. Raeisi, M. Khazaei, P. Croce, G. Tamburro, S. Comani, and F. Zappasodi, “A graph convolutional neural network for the automated detection of seizures in the neonatal EEG,” *Computer Methods and Programs in Biomedicine*, vol. 222, p. 106 950, Jul. 2022, ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2022.106950.
- [86] Z. Tong, Y. Liang, C. Sun, X. Li, D. Rosenblum, and A. Lim, “Digraph Inception Convolutional Networks,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 17 907–17 918.
- [87] C. J. Stam, G. Nolte, and A. Daffertshofer, “Phase lag index: Assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources,” *eng, Human Brain Mapping*, vol. 28, no. 11, pp. 1178–1193, Nov. 2007, ISSN: 1065-9471. DOI: 10.1002/hbm.20346.
- [88] *Mne\_connectivity.spectral.time – MNE-Connectivity 0.7.0 documentation*. [Online]. Available: [https://mne.tools/mne-connectivity/stable/\\_modules/mne\\_connectivity/spectral/time.html#spectral\\_connectivity\\_time](https://mne.tools/mne-connectivity/stable/_modules/mne_connectivity/spectral/time.html#spectral_connectivity_time) (visited on 03/28/2025).
- [89] E. S. May *et al.*, “Prefrontal gamma oscillations reflect ongoing pain intensity in chronic back pain patients,” *Human Brain Mapping*, vol. 40, no. 1, pp. 293–305, Sep. 2018, ISSN: 1065-9471. DOI: 10.1002/hbm.24373.
- [90] M. M. Nickel *et al.*, “Neural oscillations and connectivity characterizing the state of tonic experimental pain in humans,” *Human Brain Mapping*, vol. 41, no. 1, pp. 17–29, Sep. 2019, ISSN: 1065-9471. DOI: 10.1002/hbm.24784.
- [91] M. Tobe and S. Nobukawa, “Functional Connectivity Estimated using the Phase Lag Index and Transfer Entropy,” in *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Sep. 2021, pp. 1082–1087.
- [92] T. N. Kipf and M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, en, Feb. 2017.

- [93] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, *Spectral Networks and Locally Connected Networks on Graphs*, en, May 2014.
- [94] V. S. Shigehalli and V. M. Shettar, "Spectral Technique using Normalized Adjacency Matrices for Graph Matching," en,
- [95] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.6980.
- [96] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proceedings of the 14th International Conference on Neural Information Processing Systems*, ser. NIPS'00, Cambridge, MA, USA: MIT Press, Jan. 2000, pp. 381–387. (visited on 04/03/2025).
- [97] P. Lin, G. Zhu, X. Xu, Z. Wang, X. Li, and B. Li, "Brain network analysis of working memory in schizophrenia based on multi graph attention network," *Brain Research*, vol. 1831, p. 148 816, May 2024, ISSN: 0006-8993. DOI: 10.1016/j.brainres.2024.148816.
- [98] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, ISSN: 1533-7928.
- [99] L. Li *et al.*, *A System for Massively Parallel Hyperparameter Tuning*, arXiv:1810.05934 [cs], Mar. 2020. DOI: 10.48550/arXiv.1810.05934.
- [100] *Ray.tune.schedulers.AsyncHyperBandScheduler – Ray 2.44.1*. [Online]. Available: <https://docs.ray.io/en/latest/tune/api/doc/ray.tune.schedulers.AsyncHyperBandScheduler.html> (visited on 03/28/2025).
- [101] *19.5. Asynchronous Successive Halving – Dive into Deep Learning 1.0.3 documentation*. [Online]. Available: [https://d2l.ai/chapter\\_hyperparameter\\_optimization/sh\\_async.html](https://d2l.ai/chapter_hyperparameter_optimization/sh_async.html) (visited on 04/03/2025).
- [102] D. M. W. Powers, *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation*, en, Oct. 2020. (visited on 04/15/2025).
- [103] Ş. K. Çorbacioğlu and G. Aksel, "Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value," eng, *Turkish Journal of Emergency Medicine*, vol. 23, no. 4, pp. 195–198, 2023, ISSN: 2452-2473. DOI: 10.4103/tjem.tjem\_182\_23.
- [104] W. Klimesch, "The frequency architecture of brain and brain body oscillations: An analysis," *The European Journal of Neuroscience*, vol. 48, no. 7, pp. 2431–2453, Oct. 2018, ISSN: 0953-816X. DOI: 10.1111/ejn.14192.
- [105] D. J. Curwiel, "Implementing explainability in a Graph Neural Network (GNN) that analyses magnetoencephalography (MEG) data during spinal cord stimulation (SCS) in chronic pain patients to locate key brain regions used for processing neuromodulation," Tech. Rep., 2025.
- [106] D. Klepl, F. He, M. Wu, D. J. Blackburn, and P. Sarrigiannis, "Adaptive Gated Graph Convolutional Network for Explainable Diagnosis of Alzheimer's Disease Using EEG Data," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 3978–3987, 2023, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2023.3321634.
- [107] J. Zhu *et al.*, "EEG based depression recognition using improved graph convolutional neural network," *Computers in Biology and Medicine*, vol. 148, p. 105 815, Sep. 2022, ISSN: 0010-4825. DOI: 10.1016/j.compbimed.2022.105815.
- [108] H. E. Wang, C. G. Bénar, P. P. Quilichini, K. J. Friston, V. K. Jirsa, and C. Bernard, "A systematic framework for functional connectivity measures," English, *Frontiers in Neuroscience*, vol. 8, Dec. 2014, ISSN: 1662-453X. DOI: 10.3389/fnins.2014.00405.

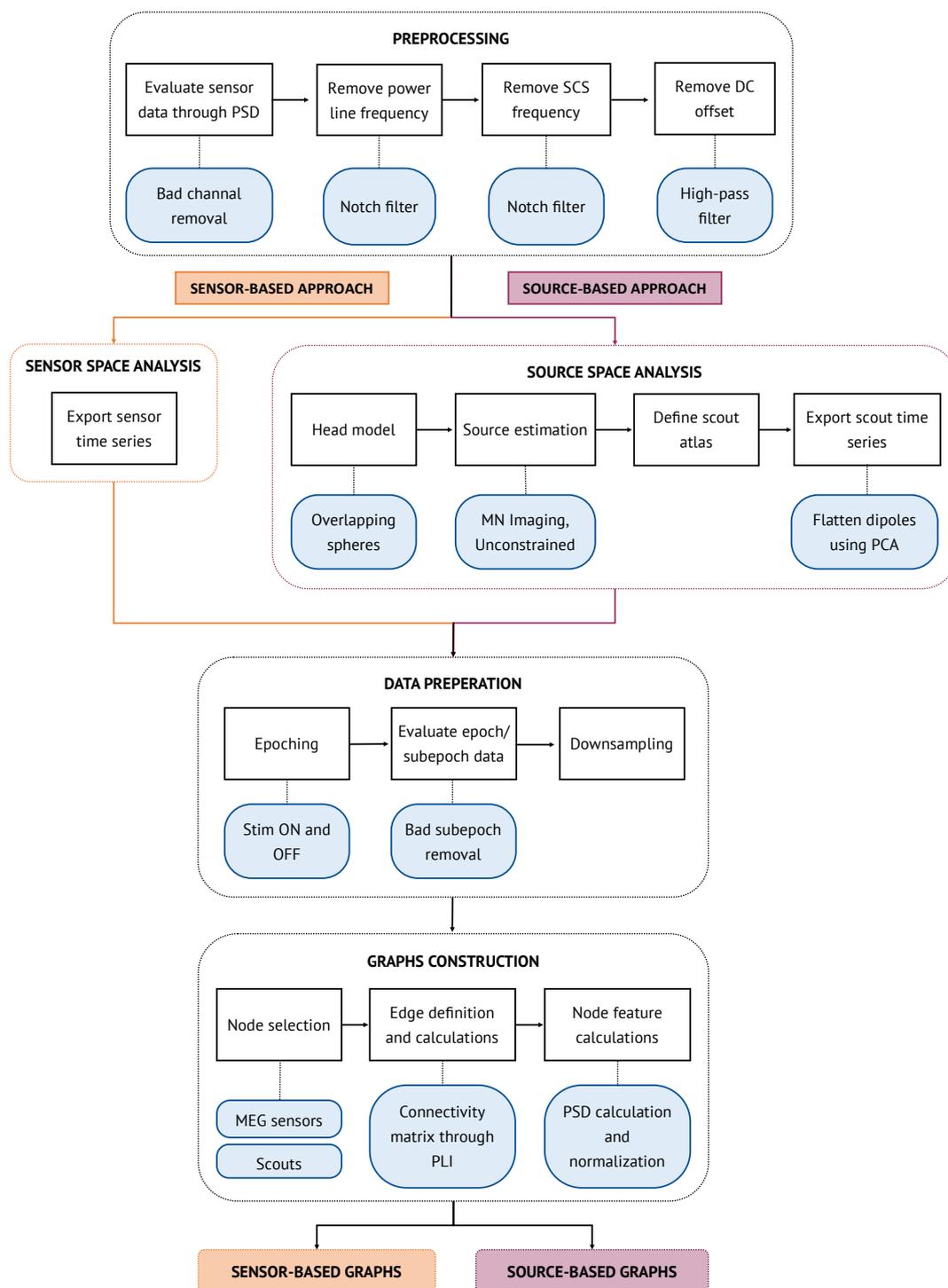
- [109] B. Kulkarni *et al.*, “Attention to pain localization and unpleasantness discriminates the functions of the medial and lateral pain systems,” *eng, The European Journal of Neuroscience*, vol. 21, no. 11, pp. 3133–3142, Jun. 2005, ISSN: 0953-816X. DOI: 10.1111/j.1460-9568.2005.04098.x.
- [110] D. De Ridder, S. Vanneste, M. Smith, and D. Adhia, “Pain and the Triple Network Model,” English, *Frontiers in Neurology*, vol. 13, Mar. 2022, Publisher: Frontiers, ISSN: 1664-2295. DOI: 10.3389/fneur.2022.757241.
- [111] M. Deogaonkar *et al.*, “Spinal Cord Stimulation (SCS) and Functional Magnetic Resonance Imaging (fMRI): Modulation of Cortical Connectivity With Therapeutic SCS,” *Neuromodulation: Technology at the Neural Interface*, vol. 19, no. 2, pp. 142–153, Feb. 2016, ISSN: 1094-7159. DOI: 10.1111/ner.12346.
- [112] K. Ueno *et al.*, “Resting-state brain functional connectivity in patients with chronic intractable pain who respond to spinal cord stimulation therapy,” *British Journal of Anaesthesia*, vol. 134, no. 2, pp. 492–500, Feb. 2025, ISSN: 0007-0912. DOI: 10.1016/j.bja.2024.10.011. (visited on 05/02/2025).
- [113] R. Dowman, D. Rissacher, and S. Schuckers, “EEG indices of tonic pain-related activity in the somatosensory cortices,” *Clinical Neurophysiology*, vol. 119, no. 5, pp. 1201–1212, May 2008, ISSN: 1388-2457. DOI: 10.1016/j.clinph.2008.01.019.
- [114] J. A. Kim and K. D. Davis, “Neural Oscillations: Understanding a Neural Code of Pain,” EN, *The Neuroscientist*, vol. 27, no. 5, pp. 544–570, Oct. 2021, Publisher: SAGE Publications Inc STM, ISSN: 1073-8584. DOI: 10.1177/1073858420958629.
- [115] L. S. Pritchep, J. Shah, H. Merkin, and E. M. Hiesiger, “Exploration of the Pathophysiology of Chronic Pain Using Quantitative EEG Source Localization,” *eng, Clinical EEG and neuroscience*, vol. 49, no. 2, pp. 103–113, Mar. 2018, ISSN: 2169-5202. DOI: 10.1177/1550059417736444.
- [116] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph Attention Networks*, en, Feb. 2018.
- [117] S. A. Holmes, A. Kim, and D. Borsook, “The brain and behavioral correlates of motor-related analgesia (MRA),” *eng, Neurobiology of Disease*, vol. 148, p. 105 158, Jan. 2021, ISSN: 1095-953X. DOI: 10.1016/j.nbd.2020.105158.
- [118] K. Lançon, C. Qu, E. Navratilova, F. Porreca, and P. Séguéla, “Decreased dopaminergic inhibition of pyramidal neurons in anterior cingulate cortex maintains chronic neuropathic pain,” *eng, Cell Reports*, vol. 37, no. 9, p. 109 933, Nov. 2021, ISSN: 2211-1247. DOI: 10.1016/j.celrep.2021.109933.
- [119] K. B. Jensen, C. Regenbogen, M. C. Ohse, J. Frasnelli, J. Freiherr, and J. N. Lundström, “Brain activations during pain: A neuroimaging meta-analysis of patients with pain and healthy controls,” *eng, Pain*, vol. 157, no. 6, pp. 1279–1286, Jun. 2016, ISSN: 1872-6623. DOI: 10.1097/j.pain.0000000000000517.
- [120] C. J. Starr *et al.*, “Roles of the Insular Cortex in the Modulation of Pain: Insights from Brain Lesions,” *The Journal of Neuroscience*, vol. 29, no. 9, pp. 2684–2694, Mar. 2009, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5173-08.2009.
- [121] A. Mayr *et al.*, “Patients with chronic pain exhibit individually unique cortical signatures of pain encoding,” *Human Brain Mapping*, vol. 43, no. 5, pp. 1676–1693, Dec. 2021, ISSN: 1065-9471. DOI: 10.1002/hbm.25750.
- [122] A. Kucyi and K. D. Davis, “The dynamic pain connectome,” *Trends in Neurosciences*, vol. 38, no. 2, pp. 86–95, Feb. 2015, ISSN: 0166-2236. DOI: 10.1016/j.tins.2014.11.006.
- [123] *Welcome to the FieldTrip website*, en. [Online]. Available: <https://www.fieldtriptoolbox.org/> (visited on 01/22/2025).
- [124] *Tutorials/TutDigitize - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/TutDigitize> (visited on 03/28/2025).
- [125] *Tutorials/TutWarping - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/TutWarping> (visited on 03/28/2025).

- [126] J. a. M. Luijten, “Magnetoencephalography to image the influence of different spinal cord stimulation paradigms on somatosensory evoked responses,” 2023.
- [127] *Tutorials/Scouts - Brainstorm*. [Online]. Available: <https://neuroimage.usc.edu/brainstorm/Tutorials/Scouts> (visited on 03/28/2025).
- [128] C. Tallon-Baudry, O. Bertrand, C. Delpuech, and J. Pernier, “Oscillatory gamma-band (30–70 Hz) Activity Induced by a Visual Search Task in Humans,” en, *Journal of Neuroscience*, vol. 17, no. 2, pp. 722–734, Jan. 1997, Publisher: Society for Neuroscience Section: Articles, ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.17-02-00722.1997.
- [129] J. Gross *et al.*, “Good practice for conducting and reporting MEG research,” *Neuroimage*, vol. 65, pp. 349–363, Jan. 2013, ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2012.10.001.
- [130] T. R. Knösche, “Transformation of whole-head MEG recordings between different sensor positions,” eng, *Biomedizinische Technik. Biomedical Engineering*, vol. 47, no. 3, pp. 59–62, Mar. 2002, ISSN: 0013-5585. DOI: 10.1515/bmte.2002.47.3.59.
- [131] M. Medvedovsky, S. Taulu, R. Bikmullina, and R. Paetau, “Artifact and head movement compensation in MEG,” eng, *Neurology, Neurophysiology, and Neuroscience*, p. 4, Oct. 2007, ISSN: 1933-1266.
- [132] M. Fraschini, M. Demuru, A. Crobe, F. Marrosu, C. J. Stam, and A. Hillebrand, “The effect of epoch length on estimated EEG functional connectivity and brain network organisation,” en, *Journal of Neural Engineering*, vol. 13, no. 3, p. 036 015, May 2016, Publisher: IOP Publishing, ISSN: 1741-2552. DOI: 10.1088/1741-2560/13/3/036015.
- [133] M. N. Vu and M. T. Thai, *PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks*, arXiv:2010.05788 [cs], Oct. 2020. DOI: 10.48550/arXiv.2010.05788.
- [134] D. Silver *et al.*, “Mastering the game of Go without human knowledge,” en, *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017, Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/nature24270.
- [135] A. E. Roth, “Introduction to the Shapley value,” in *The Shapley Value: Essays in Honor of Lloyd S. Shapley*, A. E. Roth, Ed., Cambridge: Cambridge University Press, 1988, pp. 1–28, ISBN: 978-0-521-36177-4. DOI: 10.1017/CBO9780511528446.002.
- [136] X. Zheng *et al.*, *Towards Robust Fidelity for Evaluating Explainability of Graph Neural Networks*, arXiv:2310.01820 [cs], Mar. 2024. DOI: 10.48550/arXiv.2310.01820.
- [137] Y. Mahlau, L. Kayser, and L. Berg, “[Re] On Explainability of Graph Neural Networks via Subgraph Explorations,” *ReScience C*, vol. 9, no. 2, #41, Jul. 2023. DOI: 10.5281/zenodo.8173753.

# | Appendices

# A | Data Processing Pipeline

Figure A.1 shows an overview of the graph dataset construction process, as described in the Methods section of this thesis.



**Figure A.1:** Flowchart illustrating an overview of the steps for creating the graph dataset.

DC: Direct Current; MEG: Magnetoencephalography; MN: Minimum Norm; PCA: Principle Component Analysis; PLI: Phase Lag Index; PSD: Power Spectral Density; SCS: Spinal Cord Stimulation; Stim ON and OFF: Stimulation ON and OFF.

# B | Brainstorm and Python Integration

Since more documentation has been published on creating graphs and developing Graph Neural Network (GNN) models using Python, I decided to utilize Python for further data processing steps. In order to make use of the filtered sensor space and source space data from Brainstorm, an integration step was necessary. This integration step implemented the Fieldtrip toolbox in Matlab and the MNE-Python library in Python [68, 123]. The MNE-Python library is a toolbox for processing, exploring, visualizing, and analysing electrophysiological data, such as Magnetoencephalography (MEG), and is largely compatible with data in a Functional Imaging File (fif-file) format. Furthermore, MNE-Python has functionality to extract channel time series from FieldTrip structures.

## B.1. Sensor Space Data

For the sensor space analysis, I converted the filtered MEG time series into Fieldtrip structures using Brainstorm functionality. Using MNE-Python, I created fif-files based on these data structures for further data analysis steps in Python. The fif-files also incorporate an information structure that is compatible with MNE-Python. This information structure holds the MEG channel names, the sampling frequency, and the nominal channel positions, which are useful for further analysis steps and data visualization.

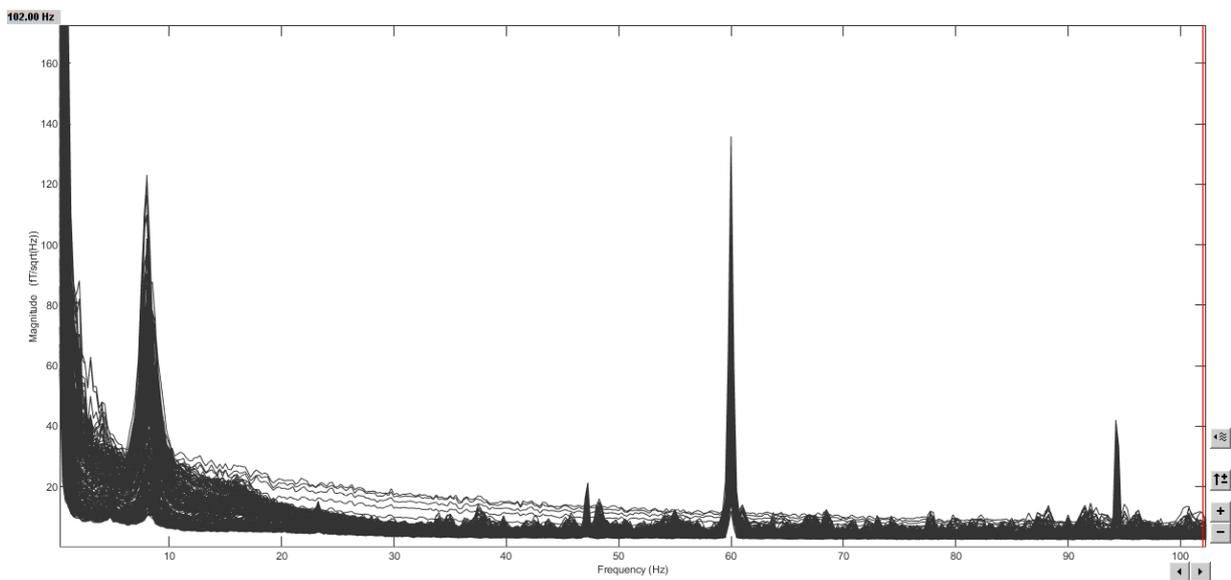
## B.2. Source Space Data

After source estimation, I exported the source space data from Brainstorm as Matlab structures. I processed these structures in Python and saved the source space data with the corresponding scouts into a Python-compatible file format. To facilitate further analysis steps, the name of this file is matched to the corresponding fif-file, containing the information structure.

# C | Preprocessing Steps

## C.1. Data Cleaning

The Magnetoencephalography (MEG) data cleaning steps were decided by inspecting the Power Spectral Density (PSD) plot for each patient file, using Welch's method with a window of four seconds and 50% overlap. An example of a PSD plot that is inspected during preprocessing is shown in Figure C.1. This plot shows the stimulation artifact at 47 Hz, with a harmonic frequency at 94 Hz, and the power line artifact at 60 Hz. Furthermore, this plot shows several bad channels, which were identified and removed.



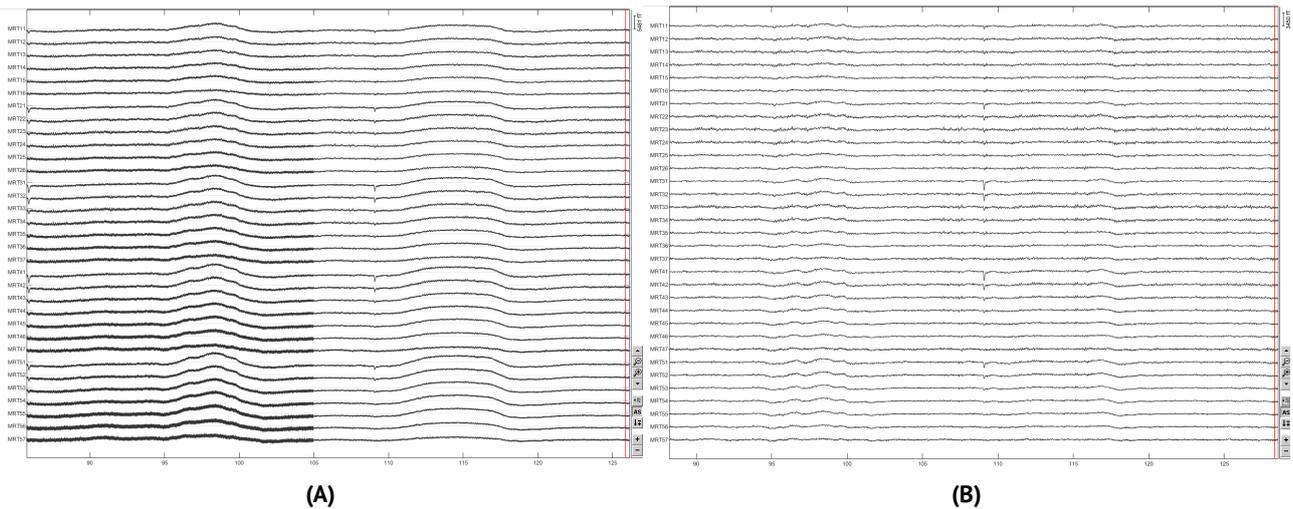
**Figure C.1:** A PSD plot used for inspecting the data.

### C.1.1. Notch Filters

Since both power line noise and Spinal Cord Stimulation (SCS) signals operate at constant frequencies, notch filters were applied to remove these sources of contamination. Power line contamination was removed through a zero-phase lag, second-order Infinite Impulse Response (IIR) notch filter with a 2 dB notch width. The specific frequencies targeted depended on the recording location: 50 Hz, 100 Hz, 150 Hz, and 200 Hz for recordings at the Donders Institute (the Netherlands), and 60 Hz, 120 Hz, and 180 Hz for recordings at the Montreal Neurological Institute (Canada). The stimulation frequency of the spinal cord stimulator was removed with the same notch filter at the frequency that was assessed in the PSD plot. Plots of the MEG recordings before and after notch filtering are shown in Figure C.2.

### C.1.2. Bad Channel Removal

MEG sensors with poor signal quality were removed. Bad channels were identified through the PSD plot and the corresponding 2D topography [71]. An example of this is provided in Figure C.1, where five bad channels were identified in the PSD plot. This step is crucial for further data analysis, as poor signal quality can ultimately distort the results.



**Figure C.2:** A 40-second section of a MEG recording, showing a subset of MEG channels. (A) Before applying notch filters, the stimulation artifact is visible in the data. (B) The same section of the recording has been filtered to remove contaminations.

### C.1.3. Additional Artifact Removal

Unlike power line and SCS artifacts, physiological artifacts such as heartbeats, eye blinks, and movement are naturally present in all recordings. Because the MEG data contains epochs with stimulation ON and OFF, these artifacts are assumed to be equally distributed across both classes. This means that the Graph Neural Network (GNN) model is unlikely to learn to differentiate the classes based on these artifacts, reducing the need for additional artifact removal. Removing these components could risk altering the neural data in a way that affects the model's ability to capture meaningful patterns [72].

### C.1.4. Exclusion of Bad Segments

Bad segments were not removed in this preprocessing stage because the cyclic SCS settings dictated the timing for creating epochs. Removing segments at this stage could disrupt the epoching process and create inconsistencies in data structures. Instead, bad segments were identified and excluded in later analysis steps, ensuring that the epoching process based on the stimulation cycles remained intact, while still addressing low-quality data points before inputting in the GNN model.

## C.2. Head Model and Source Estimation

### C.2.1. Anatomy

Due to the incompatibility of SCS systems with Magnetic Resonance Imaging (MRI) scanners, individual MRI scans could not be obtained for the patients in this study. Instead, the ICBM152 template MRI anatomy, available in Brainstorm, was used for all patients.

The head shapes of the patients were digitized using the Polhemus system [124] at the time of the MEG acquisition. This system captures the positions of head localization coils, three anatomical landmarks (nasion, left preauricular point, and right preauricular point) for alignment with the default anatomy, and additional scalp points that define the subject's head shape. These digitized head points, collected before MEG acquisition, accurately represent the individual's head geometry and can be used to scale and deform the template MRI to match the individual head shape, creating a pseudo-individual anatomy [125].

For a previous study with a different research focus, the template anatomy of each patient was warped

(deformed) based on the individual digitized head shapes [126]. Since this previous study utilized MEG recordings for the same patient group during the same recording session, I decided to utilize the existing warped anatomy for my preprocessing steps to improve anatomical alignment.

### C.2.2. Source Space

In MEG recordings, variations in channel files and sensor locations prevent averaging across patients or sessions, as sensors do not correspond to the same brain regions [73, 74]. To analyse brain activity, a model is needed to map neural electric currents (source space) to the magnetic fields detected by external sensors (sensor space), considering the different tissues between the brain and MEG sensors [75]. This process, called forward modelling or solving the forward problem, produces a head model that approximates head geometry based on the subject's anatomy and sensor locations. Source estimation, the inverse problem, uses this model to estimate brain activity at multiple brain locations from the fewer sensor locations [73]. Given the ill-posed nature of this problem, multiple solutions may explain the data equally well.

### C.2.3. Surface Model

For surface modelling, the subject's cortex is downsampled to approximately 15,000 vertices to balance anatomical accuracy with computational efficiency. The overlapping spheres model, which simplifies head geometry as overlapping spheres representing the skull under each sensor, is the recommended model for MEG data [75]. This model assumes magnetic fields are less sensitive to tissue heterogeneity, enabling efficient and reliable computation.

### C.2.4. Source Estimation

For source estimation, the Minimum Norm (MN) imaging approach is used. This method is recommended because it is a simple and robust approach that is less sensitive to approximations of the head model compared to other methods [73]. MN imaging requires a noise covariance matrix to specify the noise statistics. This is provided by calculating the noise covariance using the empty-room recordings. The current at each vertex is modelled by the orientation of an equivalent current dipole. For surface source estimation, the orientation of these dipoles can be set to be normal to the cortex (constrained) or can consist of three orthogonal dipoles along the Cartesian directions of the coordinate system for each vertex (unconstrained). The unconstrained option is recommended when using the template anatomy instead of individual MRI scans, as this may account for some of the model uncertainties [73]. This approach results in three time series, displaying the activity for each vertex in three orientations.

### C.2.5. Scout Time Series

The three time series for each vertex, representing different dipole orientations, were combined into a single representative time series using Principle Component Analysis (PCA). PCA was applied to reduce dimensionality while retaining the most relevant signal characteristics. This ensures that the extracted time series best represents neural activity at each location by minimizing the influence of noise and redundant information in the orthogonal orientations.

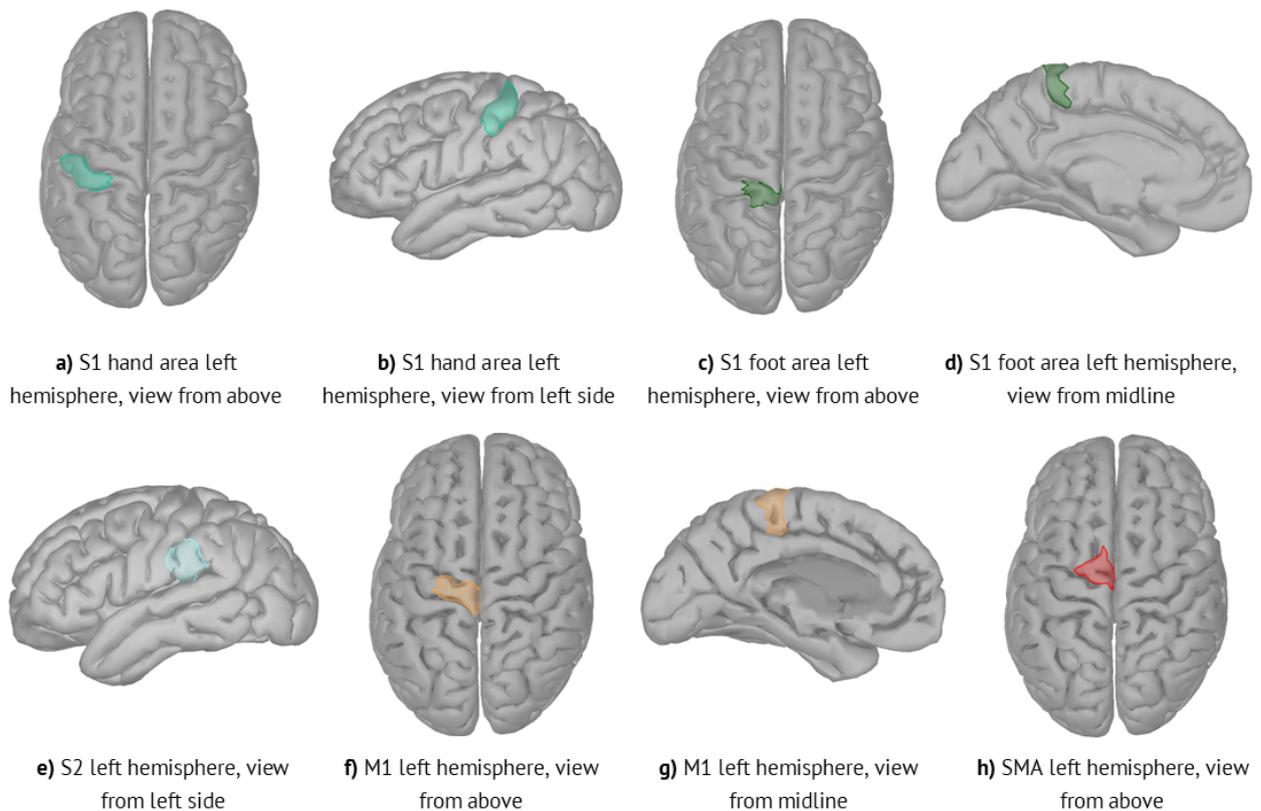
Scout time series extraction was performed following Brainstorm's recommended workflow [127]. A predefined atlas was used to define scouts, which correspond to regions of interest listed in Appendix D. Each scout consists of multiple vertices, and the source activity within each scout is averaged over all included vertices to produce a representative time series for each region of interest. The extracted scout time series data were subsequently exported for further analysis in Python.

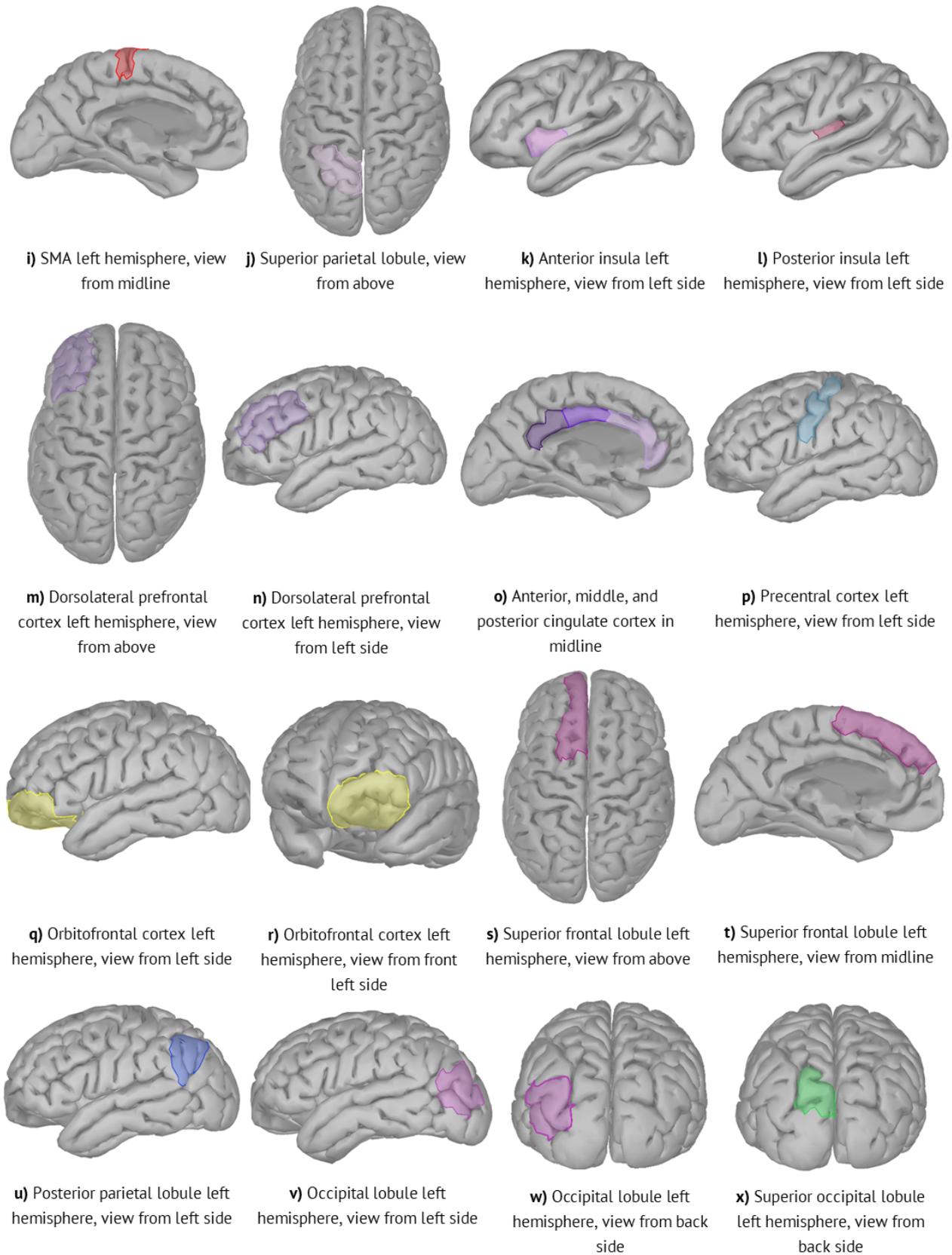
# D | Regions of Interest

The scouts were selected to include regions associated with the lateral ascending pathway (sensory-discriminative component) and the medial ascending pathway (affective-motivational component) of pain. The brain regions associated with the sensory-discriminative aspect of pain were: the primary somatosensory cortex (S1) area related to the foot, the S1 related to the hand, the primary motor cortex (M1), the secondary somatosensory cortex (S2), the supplementary motor area (SMA) and the superior parietal lobule. All scouts were created for the left and right hemispheres.

The brain regions associated with the affective-motivational component of pain were: the anterior and posterior insular cortices, anterior cingulate cortex (ACC), middle cingulate cortex (MCC), posterior cingulate cortex (PCC), dorsolateral prefrontal cortex (DLPFC) and the orbitofrontal cortex (OFC). Similarly, all scouts were created for both hemispheres. The left and right scouts for the cingulate cortices were combined, since the left and right cingulate cortices are anatomically very close to each other.

In addition to pain-related regions, scouts were also defined for broader cortical areas, including regions of the frontal, occipital, and occipito-temporal lobes, to provide a more comprehensive view of whole-brain network dynamics. This allows for a graph-based analysis that captures connectivity across distributed neural networks. All scouts are shown in Figure D.1





**Figure D.1:** Overview of the scouts defined on the cortical surface.

*M1: primary motor cortex; SMA: supplementary motor area; S1: primary somatosensory cortex; S2: secondary somatosensory cortex.*

## D.1. Scout Acronyms

A list of all scout names, as well as the corresponding acronyms and indices that are used in this thesis, is presented below.

**Table D.1:** Overview of all scout names and corresponding acronyms and indices.

Scout Name	Scout Acronym	Index
Primary Somatosensory Cortex, hand area, left	S1 Hand L	0
Primary Somatosensory Cortex, hand area, right	S1 Hand R	21
Primary Somatosensory Cortex, foot area, left	S1 Foot L	1
Primary Somatosensory Cortex, foot area, right	S1 Foot R	4
Secondary Somatosensory cortex left	S2 L	2
Secondary Somatosensory cortex right	S2 R	22
Anterior insular cortex left	ant. Insula L	3
Anterior insular cortex right	ant. Insula R	27
Posterior insular cortex left	Insula post. L	5
Posterior insular cortex right	post. Insula R	29
Primary Motor Cortex left	M1 L	6
Primary Motor Cortex right	M1 R	24
Supplementary Motor Area left	SMA L	7
Supplementary Motor Area right	SMA R	23
Superior Frontal lobe left	sup. Frontal L	9
Superior Frontal lobe right	sup. Frontal R	20
Dorsolateral Prefrontal Cortex left	DLPFC L	10
Dorsolateral Prefrontal Cortex right	DLPFC R	15
Posterior Parietal lobe left	post. Parietal L	11
Posterior Parietal lobe right	post. Parietal R	19
Superior Occipital lobe left	sup. Occipital L	12
Superior Occipital lobe right	sup. Occipital R	17
Occipital lobe left	Occipital L	13
Occipital lobe right	Occipital R	18
Precentral cortex left	Precentral L	14
Precentral cortex right	Precentral R	16
Orbitofrontal cortex left	OFC L	32
Orbitofrontal cortex right	OFC R	30
Posterior Cingulate Cortex	PCC	25
Middle Cingulate Cortex	MCC	26
Anterior Cingulate Cortex	ACC	31
Superior Parietal lobe left	sup. Parietal L	8
Superior Parietal lobe right	sup. Parietal R	28

# E | Subepoch Duration and Overlap

For finding the optimal subepoch duration and overlap, a subanalysis was performed. Since there is no existing literature on the optimal subepoch length for Magnetoencephalography (MEG) data for inputting in a Graph Neural Network (GNN) model, this optimal setting was found empirically and through analysing the outcomes of different configurations.

In this subanalysis, the filtered sensor-based MEG data was segmented into epochs and, subsequently, these epochs were split into subepochs with different configurations for duration and overlap. The resulting subepochs for each configuration were used as input for the GNN model. The model was trained and tested for each subepoch configuration, and the outcomes were evaluated. Specifically, the accuracy of the test set was reported to assess the effect of the different subepoch configurations.

Four different configurations of subepoch duration and overlap were evaluated, as provided in Table E.1. Each configuration of duration and overlap results in a different number of subepochs. For this subanalysis, all patient files, including both tonic and burst stimulation paradigms, were utilized. The number of subepochs represents the amount of subepochs used for graph creation, which are subsequently used for model training and testing. The number of bad subepochs is the amount of subepochs that are labelled as bad using the bad subepoch detection approach (Appendix F) before the graphs are created.

**Table E.1:** Different configurations for subepoch duration and overlap, the resulting number of subepochs for ultimate graph creation, and the number of bad subepochs.

Duration (s)	Overlap (s)	Number of subepochs	Number of bad subepochs
30	5	187	3
10	5	1683	16
30	25	930	20
15	10	1501	19

A shorter subepoch duration increases the number of resulting subepochs, thereby augmenting the graph dataset. Similarly, applying a large overlap between subepochs further increases the total number of subepochs. However, a larger overlap also elevates the number of subepochs marked as bad, since overlapping the segments of MEG data containing noise or artifacts will lead to more segments with noise or artifacts. Given the relatively small number of patients included in this study, increasing the number of subepochs through overlapping was advantageous for expanding the graph dataset and improving model robustness.

Additionally, longer subepochs were preferred, as they produce smoother Power Spectral Density (PSD) estimates, by increasing the number of windows that can be used for PSD computation. Moreover, subepochs must be sufficiently long to support Phase Lag Index (PLI) computations, which rely on time-frequency decomposition using Morlet wavelets [88]. These wavelets extract frequency-specific phase information, enabling the calculation of phase differences between signals, which is a critical step in deriving the PLI. The duration of a Morlet wavelet depends on the frequency of interest and the number of cycles used. With the default setting of seven cycles, the wavelet for 1 Hz spans seven seconds. Therefore, selecting subepochs shorter than this duration would compromise the reliability of PLI estimates and is not recommended [128].

The accuracy of the test set for each subepoch configuration is reported in Table E.2. The accuracies ranged from 0.81 to 0.98, with the highest performance achieved using 30-second subepochs and a 25-second overlap.

**Table E.2:** Test accuracy for the different configurations for subepoch duration and overlap.

<b>Duration (s)</b>	<b>Overlap (s)</b>	<b>Test accuracy</b>
30	5	0.95
10	5	0.81
30	25	0.98
15	10	0.91

Since a broader analysis was not feasible for this study due to time constraints, I selected one configuration as the most optimal for this study and used this subepoch configuration in all subsequent steps. This decision was based on balancing the number of subepochs created, the number of bad subepochs detected, and the overall test accuracy across configurations. Hence, I chose a subepoch duration of 30 seconds with a 25-second overlap as the most optimal configuration.

# F | Bad Subepoch Detection

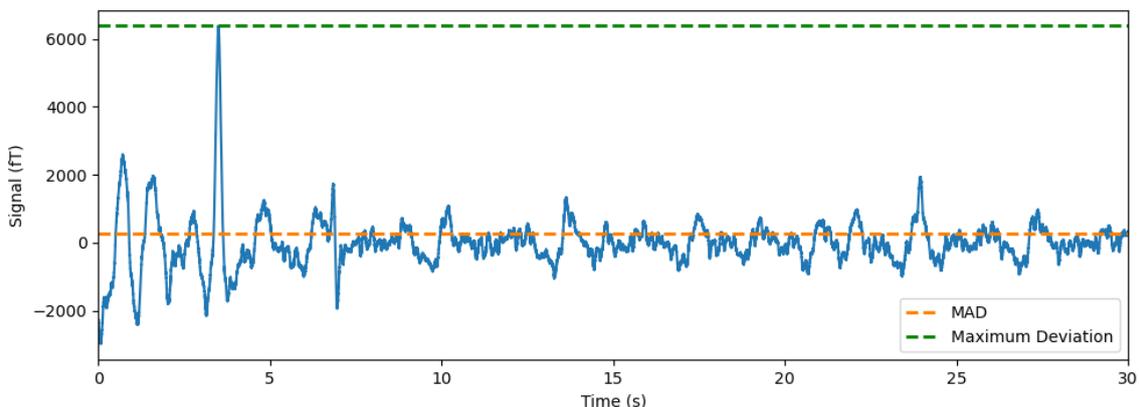
For detection of noise and artifacts in the subepochs, a method is implemented based on statistical measures of the time series and a threshold. Since this detection method may be largely affected by the subepoch length, as the artifacts may be captured in the entirety of the 30-second subepoch, this method uses statistical measures of an entire epoch (50-second window of Magnetoencephalography (MEG) data), as well as measures that assess the deviation of the subepoch with respect to the epoch.

First, the median of the time series data within one epoch is computed. The median is used because of its ability to accurately represent a typical value of the epoch data, as it is less influenced by outliers compared to the mean or standard deviation. Subsequently, the Median Absolute Difference (MAD) is calculated for one epoch, which is a robust measure of variability or spread of the data. It is computed by finding the median of the absolute deviations between each data point in the epoch and the median value of this epoch. The MAD describes how much data typically varies around the median of the signal.

After this, the maximum deviation of the subepoch is determined, entailing the absolute largest distance between any data point in the subepoch and the median of the data. From these values, a deviation score is retrieved using Equation F.1 that highlights how extreme the maximum deviation is compared to the typical spread of the data, as described by the MAD.

$$\text{deviation score} = \frac{\text{maximum deviation of subepoch}}{\text{MAD of epoch}} \quad (\text{F.1})$$

A large deviation score suggests that the maximum deviation is much larger than what would be expected based on the typical data spread, indicating the presence of potential artifacts. A threshold is defined for this deviation score, and a subepoch is marked as 'bad' and removed if the deviation score exceeds this threshold. This threshold is established based on deviation scores from various subepochs, and is set to 100 for the sensor-based time series, and 300 for the source-based scout data. This threshold aims to effectively remove bad segments, while minimizing the loss of subepoch data. An example of a detected bad subepoch is shown in Figure F.1.

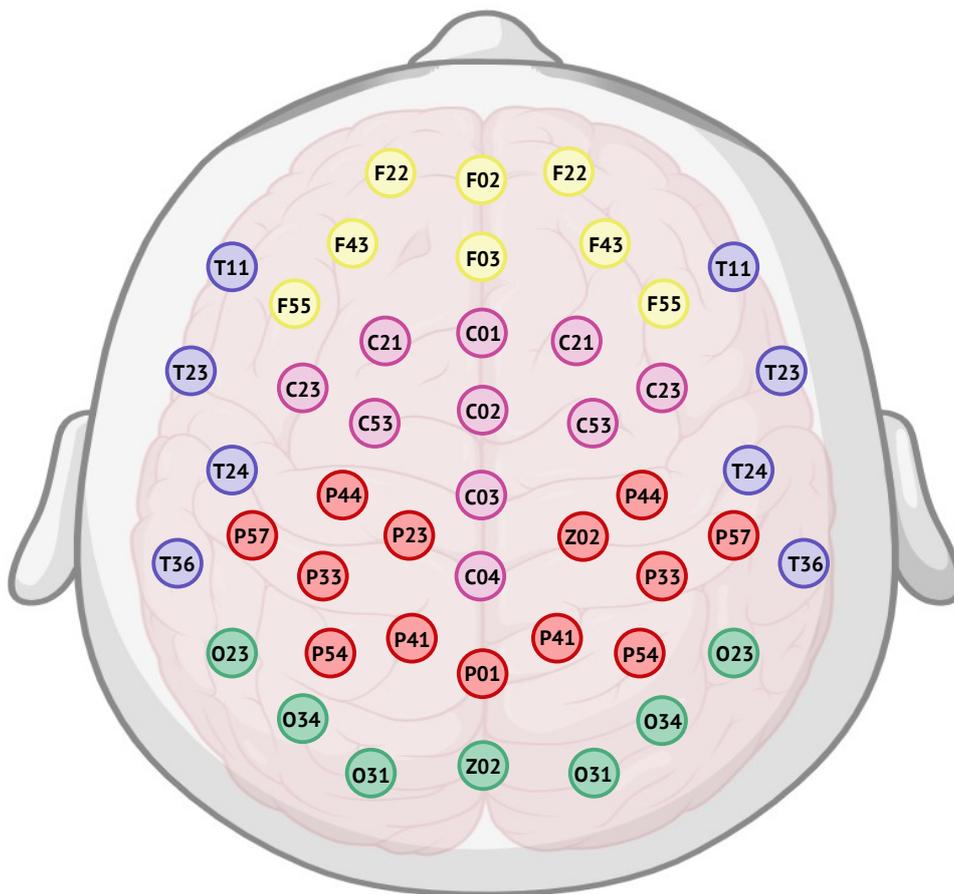


**Figure F.1:** Plot of the time series of a bad subepoch with deviation score of 122, showing the Median Absolute Difference (MAD) of the 50-second epoch and maximum deviation of this subepoch.

# G | MEG Channel Selection

Figure G.1 shows an illustration of the selected Magnetoencephalography (MEG) channels for the sensor-space analysis in this study. A total of 46 MEG channels were selected.

The names of the MEG channels are represented by a combination of letters and numbers. The first and second letters are "ML", "MR", or "MZ", corresponding to the location of the sensors (ML: left hemisphere, MR: right hemisphere, or MZ: midline). These letters are omitted in the figure for clarity. The third letter corresponds to the region of the sensor (F: frontal, C: central, P: parietal, O: occipital, and T: temporal), and the numbering is done via rows and columns.



**Figure G.1:** The Magnetoencephalography (MEG) channels that are selected for the sensor-space analysis pathway. The 46 channels are gathered based on clinical relevant brain areas. Each colour in the illustration is representative of a brain lobe (yellow: frontal, magenta: central, red: parietal, purple: temporal, and green: occipital).

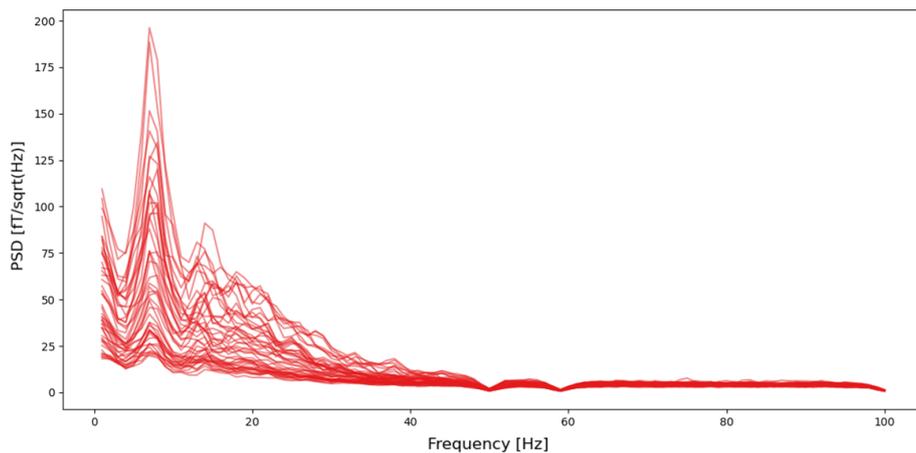
A list of all MEG channel names, as well as the corresponding indices that are used in this thesis, is presented below.

**Table G.1:** Overview of Magnetoencephalography (MEG) channel names and corresponding indices.

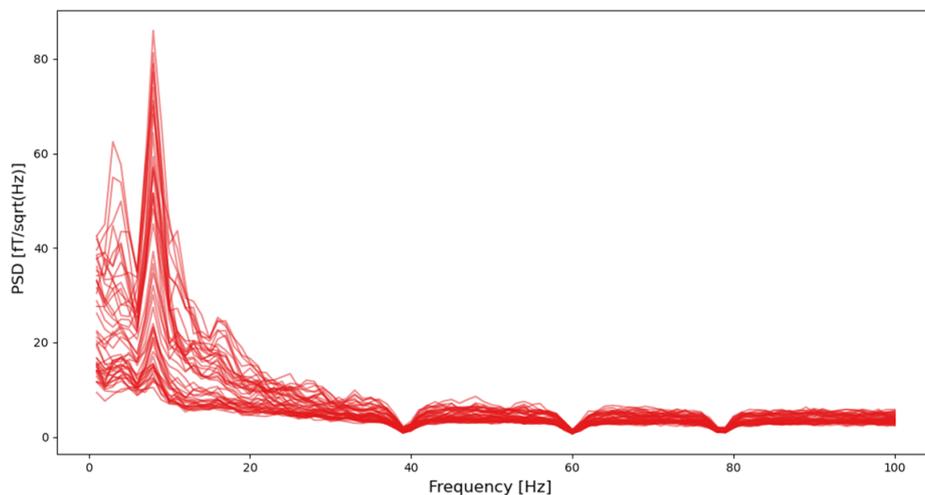
<b>MEG Channel Name</b>	<b>Index</b>	<b>MEG Channel Name</b>	<b>Index</b>
MZO02	3	MLT23	8
MZP01	41	MRT23	20
MZC04	45	MLC23	28
MZC03	44	MRC23	24
MZC02	43	MLF55	13
MZC01	42	MRF55	18
MZF03	15	MLT11	7
MZF02	14	MRT11	19
MLO31	0	MLC21	27
MRO31	4	MRC21	23
MLO23	1	MLF43	12
MRO23	5	MRF43	17
MLP41	33	MLF22	11
MRP41	38	MRF22	16
MLP54	34	MLT36	9
MRP54	39	MRT36	21
MLO34	2	MLP57	35
MRO34	6	MRP57	40
MLP33	31	MLT24	10
MRP33	36	MRT24	22
MLP44	32	MLC53	29
MRP44	37	MRC53	25
MLP23	30	MRP23	26

# H | Node Feature Normalization

The Power Spectral Density (PSD) values were computed for all subepochs across all patient files. However, substantial variability in PSD values between different patient recordings posed a challenge for the GNN model, making it difficult to identify consistent patterns across multiple files. This variability likely stemmed from individual differences in brain activity, sensor placement, and recording conditions. The variability in PSD values is shown in Figure H.1 and H.2, where the PSD values for two different subepochs are plotted.



**Figure H.1:** A plot of the PSD values for one subepoch during stimulation ON. The different lines represent the 46 different nodes in the sensor-space analysis.



**Figure H.2:** A plot of the PSD values for one subepoch during stimulation ON. The different lines represent the 46 different nodes in the sensor-space analysis.

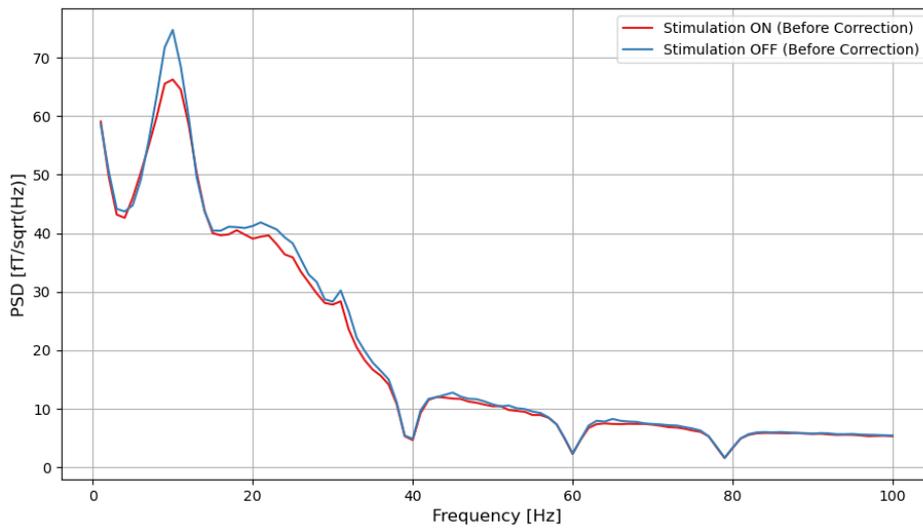
Furthermore, previous research discourages sensor-space averaging across different runs or subjects due to the potential for head movements under the Magnetoencephalography (MEG) array during recordings [129]. Such movements can cause sensors to record activity from different parts of the brain, making direct comparisons unreliable unless explicit motion compensation techniques are applied [130, 131].

To address these challenges, normalization was performed based on baseline-corrected PSD values for each patient file. Specifically, each PSD value computed for one patient was adjusted relative to the mean

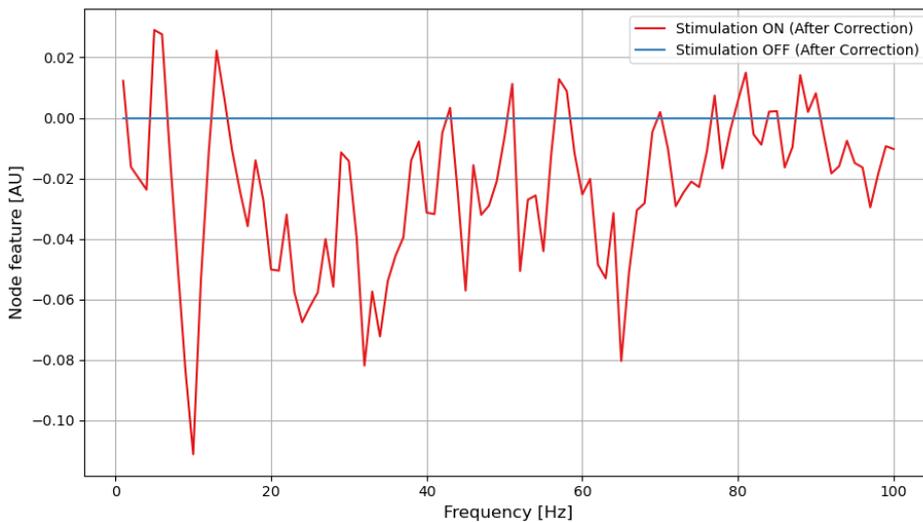
PSD of subepochs during stimulation OFF in this patient, following Equation H.1. This normalization step standardizes spectral power across files, allowing the Graph Neural Network (GNN) model to focus on meaningful neural activity patterns rather than inter-subject variability or sensor-related noise.

$$Node\ features_{norm} = \frac{PSD - PSD_{OFF,mean}}{PSD_{OFF,mean}} \quad (H.1)$$

Where the  $PSD_{OFF,mean}$  is the mean of all PSDs computed from subepochs with stimulation OFF from one file,  $PSD$  is the current PSD before normalization, entailing a value for all frequency bins, and  $Node\ features_{norm}$  refers to the node features after normalization. After normalization, the node feature matrix, consisting of node features for all nodes, comprised normalized values that highlight the differences in stimulation ON and OFF, and reduce the variability in data across multiple recordings. The average PSD values for one file before normalization are plotted in Figure H.3, and Figure H.4 shows the node features for this file after normalization.



**Figure H.3:** A plot of the average PSD values for all subepochs from one file before normalization.



**Figure H.4:** A plot of the average node features for all subepochs from one file after normalization.  
*AU: Arbitrary Units.*

# I | Phase Lag Index Computation

The Phase Lag Index (PLI) was selected as the functional connectivity metric for defining the edges in the graph representation of the Magnetoencephalography (MEG) data. The PLI is a widely used measure of phase synchronization that aims to quantify consistent phase-lead or phase-lag relationships between pairs of signals, while minimizing the effects of volume conduction and common sources [87].

The PLI was computed using the following formula [88]:

$$\text{PLI} = \left| \frac{1}{n} \sum_{t=1}^n \text{sign}(\text{Im}(S_{xy}(f, t))) \right| \quad (\text{I.1})$$

In this formula,  $S_{xy}(f, t)$  represents the cross-spectral density between signals  $x(t)$  and  $y(t)$ , at frequency  $f$  and time  $t$ . The imaginary part of this quantity,  $\text{Im}(S_{xy})$ , reflects the non-zero phase lag components. The  $\text{sign}(\cdot)$  function retains the direction of phase difference, and  $n$  refers to the number of time windows or observations over which the average is taken.

The cross-spectral density  $S_{xy}(f)$  was computed using frequency-domain analysis methods implemented in MNE-Python, with parameters selected to ensure robust estimation across the desired frequency bands. For each pair of MEG channels, the imaginary component of the cross-spectrum was extracted and passed through the sign function before averaging across the time window to compute the PLI.

A PLI value close to zero indicates either an absence of consistent phase coupling or a phase relationship centred around zero, where the signs cancel out. A PLI value approaching one indicates that one signal consistently leads or lags another with a non-zero phase difference, reflecting a stronger functional interaction. By discarding zero-phase interactions and amplitude information, the PLI avoids inflated connectivity values caused by volume conduction, making it well suited for MEG data analysis.

The selection of epoch length is critical for reliable PLI estimation. A study by Fraschini et al. [132] investigated the influence of epoch duration on the stability of PLI values across subjects. They demonstrated that epochs of 12 seconds or longer yield more stable and consistent PLI values, improving the reliability of group-level inferences. In contrast, shorter epochs, particularly those under 8 seconds, introduced greater inter-subject variability and less distinct connectivity patterns. This instability is likely due to insufficient frequency resolution and reduced reliability in phase difference estimation.

Taken this into account, and the results of the subanalysis in Appendix E, an epoch length of 30 seconds was selected in this study to ensure reliable and consistent PLI estimation across all graph instances and subjects.

# J | Node Explainability using SubgraphX

## J.1. SubgraphX explainer

The SubgraphX explainer is an advanced explainability technique specifically designed to reveal how Graph Neural Network (GNN)s make predictions by identifying and analyzing important subgraphs within the input graph [62]. Other GNN explainability methods often focus on individual nodes or edges, which can sometimes miss the larger structural patterns that are crucial for the GNN's decision-making process [64, 133]. These larger structural patterns may be responsible for influencing the model's predictions in ways that cannot be captured by focusing on individual components. SubgraphX overcomes this limitation by emphasizing the role of subgraphs, which are locally connected structures within the graph, in shaping the model's outputs.

The approach of SubgraphX involves the integration of two powerful techniques: Monte Carlo Tree Search (MCTS) and Shapley values. These methods work together to systematically explore and evaluate subgraphs within the input graph to determine which ones are most influential in the model's prediction.

### J.1.1. Monte Carlo Tree Search

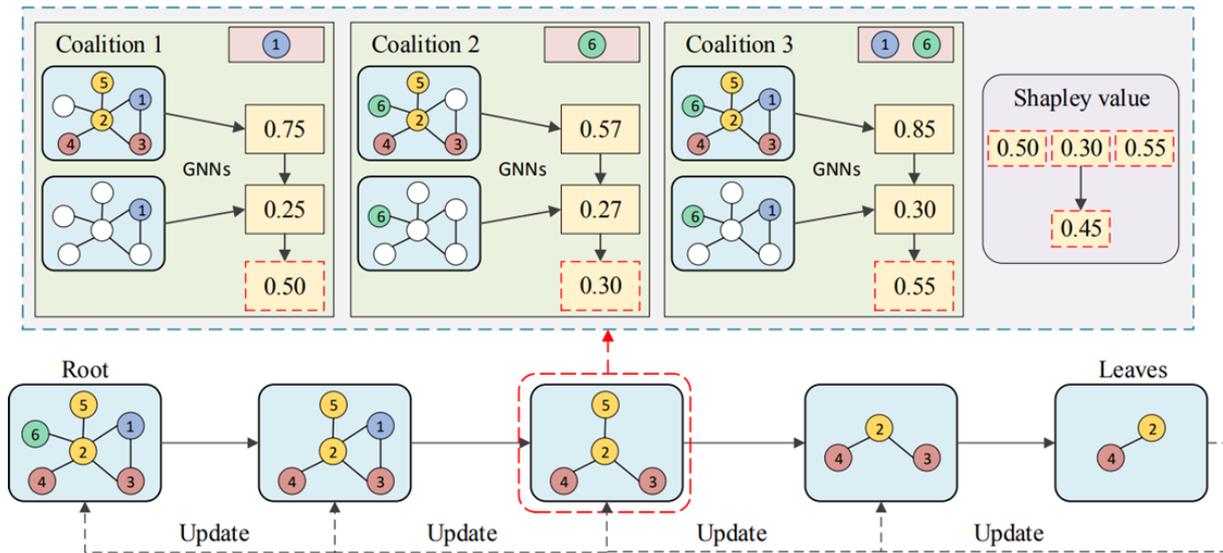
MCTS is a heuristic search algorithm widely used in decision-making tasks, particularly in environments with large and complex decision spaces [134]. The strength of MCTS lies in its ability to balance two objectives: exploration (trying new, unexplored paths) and exploitation (refining and focusing on promising paths). In the context of SubgraphX, MCTS is used to explore different subgraphs by generating multiple branching paths. The algorithm begins by selecting a set of nodes and constructing subgraphs by combining nodes together. It then systematically removes certain nodes to assess the impact of those removals on the model's prediction. This forms one search tree of the MCTS algorithm. MCTS iteratively performs two key steps:

1. Exploration: Involves creating new subgraphs that have not been tested before, allowing the algorithm to explore a wide variety of potential subgraphs.
2. Exploitation: Focuses on subgraphs that have already shown promise in improving the GNN model's predictions, refining these subgraphs to determine their significance more accurately.

### J.1.2. Shapley Values

Shapley values are a well-established method from cooperative game theory, used to fairly allocate a "payout" among participants based on their individual contributions to the overall outcome [135]. In the context of SubgraphX, the "payout" refers to the model's prediction, and the "participants" are the different subgraph structures. For each subgraph that is identified through the MCTS algorithm, Shapley values are used to measure the marginal contribution of each node outside of the subgraph. The marginal contribution refers to the difference in the GNN model's prediction when a node is added to the identified subgraph (coalition), compared to when it is not. This shows how much this specific node improves or alters the prediction of the model compared to the prediction based on the subgraph.

An illustration of the SubgraphX explainer is included in Figure J.1, where the bottom shows one selected path in the search tree, corresponding to one iteration of MCTS. For each subgraph in this search tree, the Shapley value is computed based on the marginal contributions of nodes outside of this subgraph. These marginal contributions are averaged over all coalitions to obtain a Shapley value for each subgraph. The subgraph from the MCTS search tree with the highest Shapley value is deemed the most influential in driving the GNN model's prediction.



**Figure J.1:** Illustration of the SubgraphX explainer. The bottom represents one iteration of the Monte Carlo Tree Search. For each constructed subgraph, a Shapley value is computed based on the average marginal contributions of nodes outside of this subgraph. *Source: Yuan et al. [62]*

## J.2. Fidelity and Sparsity

Fidelity and sparsity are two critical metrics used to evaluate the effectiveness of explainability methods. These metrics help ensure that the explanations provided by SubgraphX are both accurate and interpretable.

Fidelity measures the model's performance before and after the occlusion (removal) of the important subgraph that is identified through the SubgraphX explainer. Essentially, it gauges how much the GNN model's prediction changes when specific subgraphs are altered or removed. The fidelity score quantifies the difference in the GNN model's confidence in the predicted class for the normal input graph and the occluded graph, where the important subgraph is removed. A high fidelity score indicates that the occluded subgraph is crucial to the model's decision-making process, as the removal of the subgraph leads to a significant change in the confidence of the prediction.

Sparsity refers to the number of nodes used to explain a model's prediction, with respect to the total number of nodes. An explanation with high sparsity uses fewer nodes, which can make the explanation easier to understand and more interpretable. However, sparse explanations are typically less accurate because they omit potentially important nodes. Thus, achieving the right balance between fidelity and sparsity is essential for effective explainability. An ideal explanation is one that uses a minimal number of nodes (high sparsity) but still maintains high fidelity, meaning the model's performance remains similar even after removing the explained subgraph. SubgraphX is particularly advantageous in this regard, as it can achieve high fidelity without significantly increasing sparsity, making it both efficient and effective for explaining GNN predictions [62].

### J.3. Parameters

SubgraphX offers several tunable parameters that can be adjusted based on the needs of the analysis.

- **Amount of nodes:** The maximum number of nodes included in the subgraph used for analysis. Increasing the maximum number of nodes can provide a more comprehensive explanation, but it may also reduce sparsity.
- **MCTS iterations:** The number of MCTS iterations used to explore different subgraphs. More iterations generally improve the accuracy of the subgraph identification, however, this will lead to an increase in computational time.
- **Shapley iterations:** The number of iterations used to compute the Shapley values. More iterations lead to more accurate measurements of node contributions, while also increasing the computational cost.
- **Exploration and exploitation weight:** This parameter controls the balance between exploring new subgraphs and exploiting previously found subgraphs. A higher emphasis on exploration may discover new important subgraphs, while a focus on exploitation helps refine the most promising ones.
- **Classification task:** Specifies the type of task that the GNN is performing. In this study, the classification task is set to graph classification, which involves classifying entire graphs rather than individual nodes or edges.

For the implementation of SubgraphX in this study, the parameters were selected through exploration, aiming to balance a high fidelity score and the available computational resources. The selected parameters for the SubgraphX implementation are reported in Table J.1.

The number of nodes was set to be relatively small compared to the total number of nodes [105]. Zheng et al. [136] have reported that smaller subgraphs are generally more stable, though the optimal size is still uncertain. The number of MCTS iterations is preferably set to a larger value, since this promotes intensive searching for the optimal subgraph, however, this results in high computational costs. Mahlau et al. [137] reported that an increased number of Shapley iterations for application on a GNN model with convolutional layers did not significantly increase the fidelity scores, compared to lower values. Furthermore, the study by Mahlau et al. [137] explored different exploration and exploitation ratios, however, little effect on the performance was found.

**Table J.1:** Parameter settings for the SubgraphX explainability method.

Parameter	Value
Number of nodes	5
MCTS iterations	100
Shapley iterations	10
Exp. weight	0.5
Task	Graph classification

*MCTS: Monte Carlo Tree Search; Exp. weight: Exploration and exploitation weight*

# K | Patient Population

## K.1. Patient Characteristics

The patient characteristics are shown in Table K.1. The patient names were assigned for a previous study. These names were also used for the current study, however, recordings with cyclic stimulation were not made for all patients in the previous study. Therefore, the numbering of the patient names is not consistent.

**Table K.1:** Overview of the patient characteristics.

Patient	Sex	Age	PD (years)	Pain condition	Pain location	Side	Pain score tonic SCS*	Pain score burst SCS*
PT01	M	53	32	PSPS-2	Back and left leg	L	0	0
PT03	F	42	20	PSPS-2	Right hip and buttock	R	4	2
PT04	M	59	6	PSPS-2	Back, left leg and foot	L	7	5
PT05	M	52	5	PSPS-2	Right hip, buttock, leg and foot	R	6	7
PT06	F	45	16	PSPS-2	Back	R, L	1	2
PT07	M	58	31	PSPS-2	Back and left leg	L	3	2
PT08	F	42	19	PSPS-2	Back, left hip, buttock, and leg	L	4	2
PT09	F	62	12	PSPS-2	Back, neck, right buttock and leg	R	6	9
PT10	M	70	15	DNP	Both feet	R, L	6	4
PT11	F	62	20	PSPS-2	Back, right buttock and leg	R	6	6
PTN04	F	43	23	PSPS-2	Back, left leg and foot	L	4	3
PTN05	M	64	9	PSPS-2	Back	R, L	6	2
PTN06	M	70	21	PSPS-2, DNP	Right leg, buttock and foot	R	1	1
PTN07	F	56	3	CRPS	Back and left foot	L	7	7
PTN08	F	40	5	PSPS-2	Right leg and foot	R	2	5
PTN09	F	56	35	PSPS-2	Back and right leg	R	3	2
PTN10	F	49	13	PSPS-2	Left leg, buttock and foot	L	5	5
PTN11	M	63	15	DNP	Back and left leg	L	2	7
PTN12	M	38	7	NP	Left leg, knee, and foot	L	-	-
PTN13	M	53	15	PSPS-2	Back and left leg	L	4	-
PTN14	M	68	29	PSPS-2	Right back and buttock	R	2	3
PTN15	M	60	40	PSPS-2	Left back, buttock and leg	L	7	6

\* The pain scores were assessed using the Numeric Rating Scale (NRS) from 0 to 10, where 0 was no pain and 10 was the worst pain imaginable.

CRPS: Complex Regional Pain Syndrome; DNP: Diabetic Neuropathy; M/F: male/female; NP: neuropathy; PD: Pain Duration; PSPS-2: Persistent Spinal Pain Syndrome type 2; R/L: right/left; SCS: Spinal Cord Stimulation.

## K.2. Stimulation Parameters

The stimulation parameters for each patient are shown in Table K.2, describing the stimulation frequency that was selected for tonic and burst Spinal Cord Stimulation (SCS) for each patient. If the stimulation frequency could not be assessed in the Power Spectral Density (PSD) plot, the time points for the cyclic stimulation could also not be assessed, and hence, this recording was excluded.

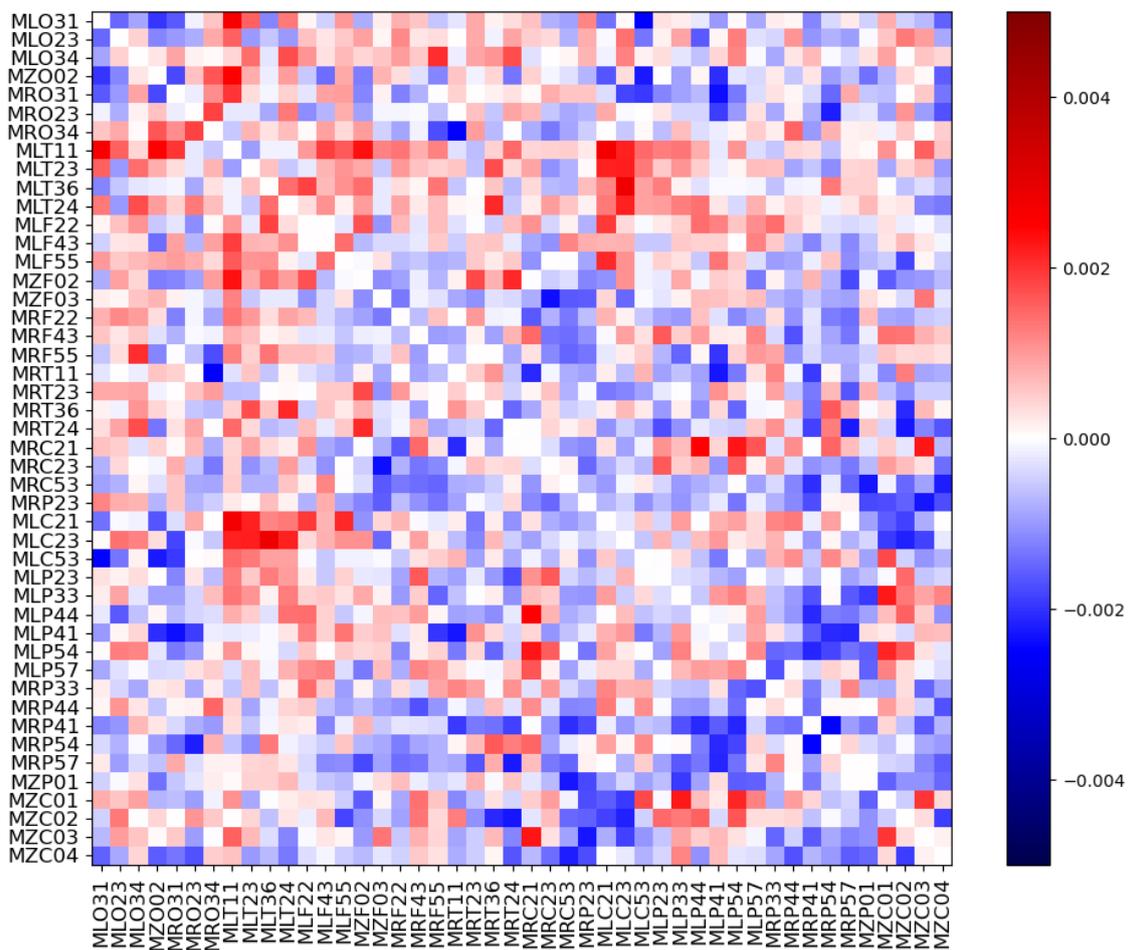
**Table K.2:** Overview of the stimulation parameters for all patients, and excluded recordings.

Patient	Burst SCS (Hz)	Tonic SCS (Hz)
PT01	40	-*
PT03	39	59
PT04	39	59
PT05	39	Excl.
PT06	39	41
PT07	51	39
PT08	39	29
PT09	39	60
PT10	39	39
PT11	39	60
PTN04	39	59
PTN05	39	59
PTN06	39	60
PTN07	39	59
PTN08	39	Excl.
PTN09	39	30
PTN10	39	Excl.
PTN11	Excl.	Excl.
PTN12	Excl.	Excl.
PTN13	Excl.	70
PTN14	39	40
PTN15	39	39

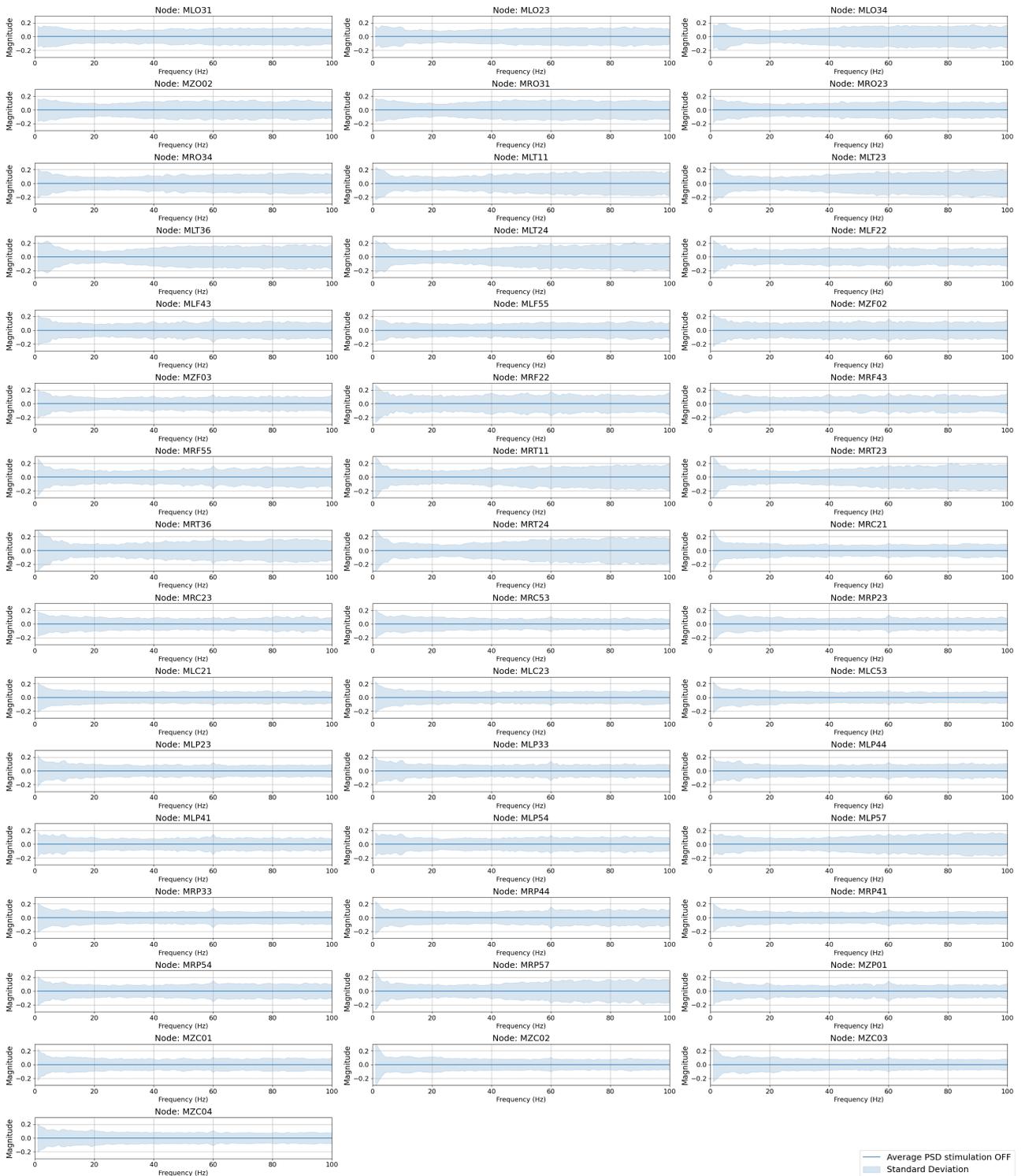
\* For this patient, the tonic stimulation recording was missing.

Excl.: excluded measurement; SCS: Spinal Cord Stimulation.

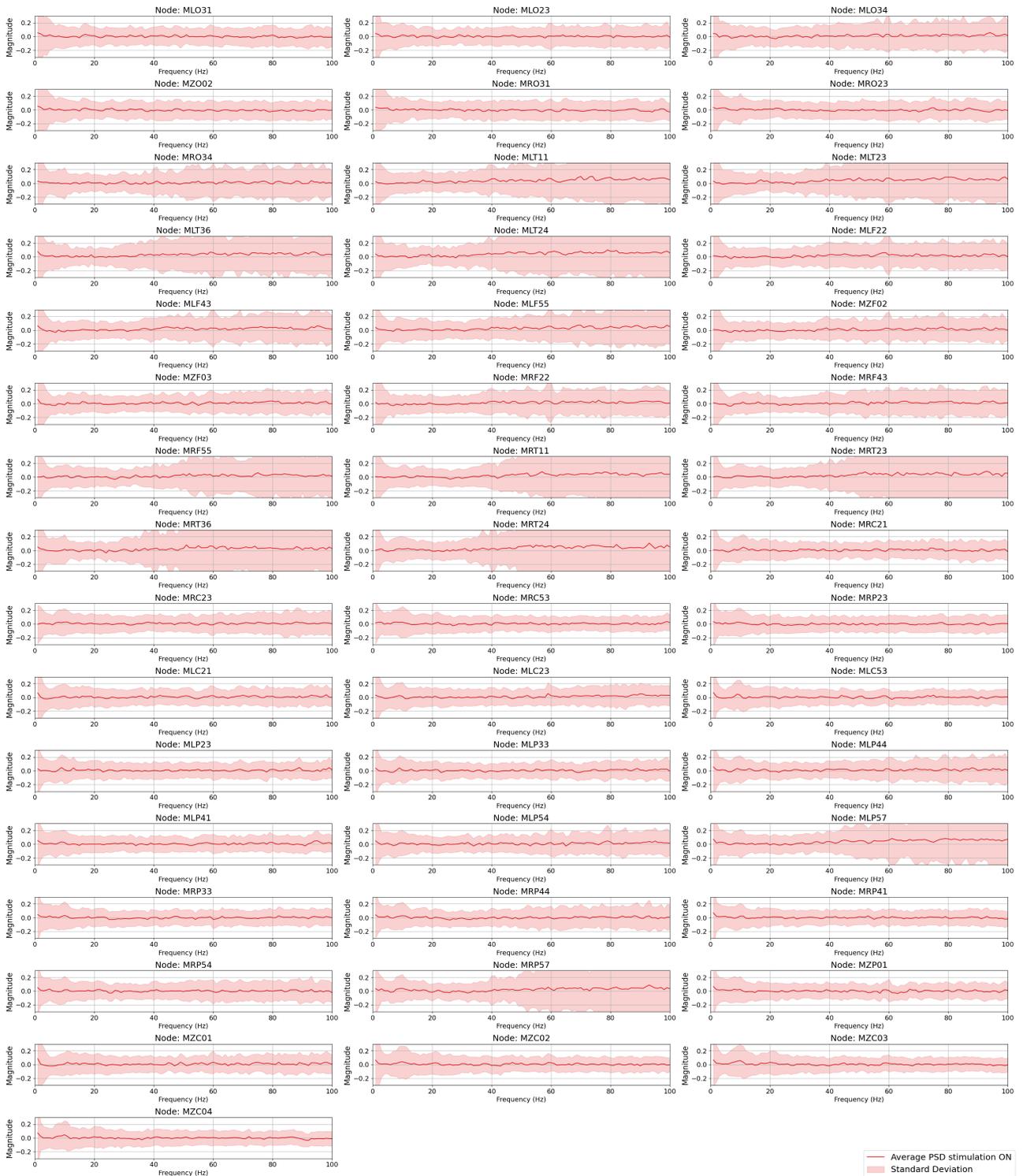
# L | Connectivity Differences and Node Features



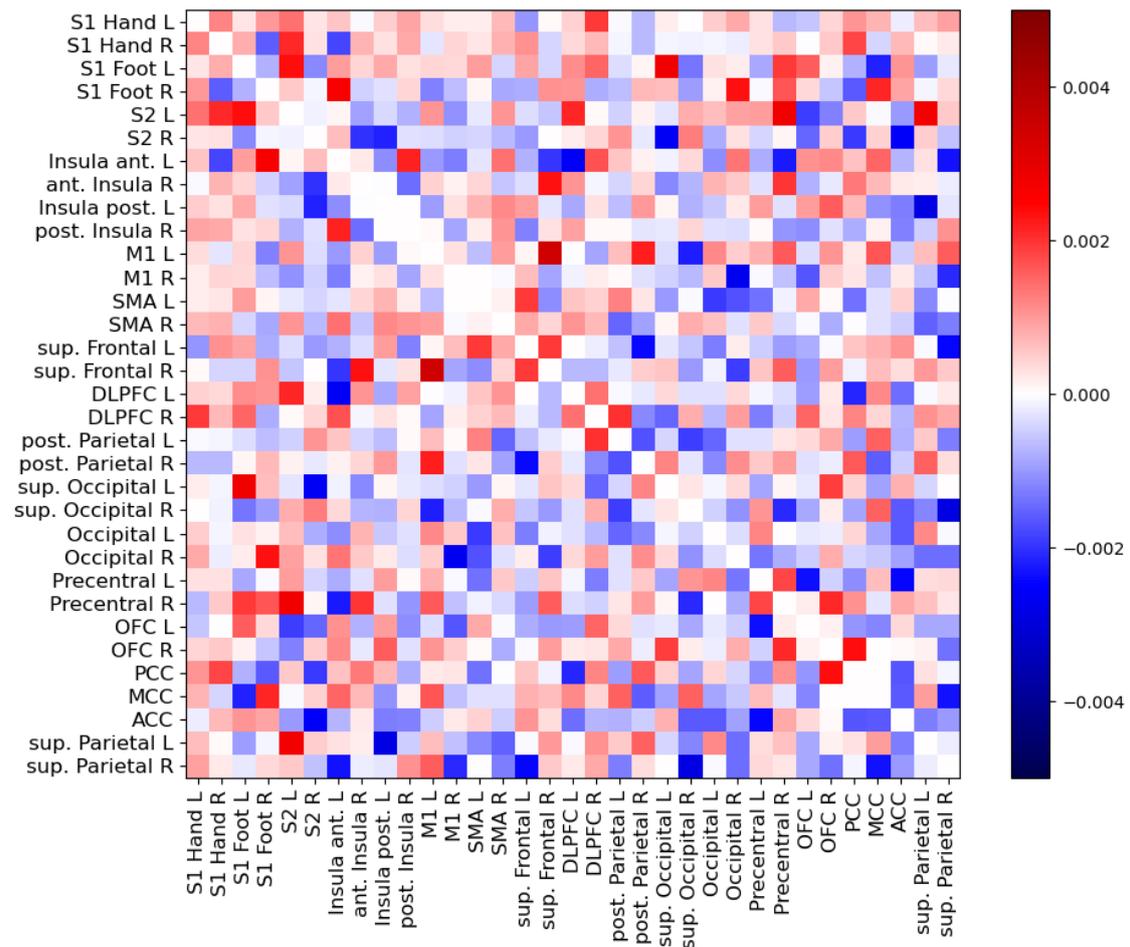
**Figure L.1:** The difference in connectivity between stimulation ON and OFF, respectively, shown as a heatmap where red indicates higher connectivity during stimulation ON, and blue indicates higher connectivity during stimulation OFF.



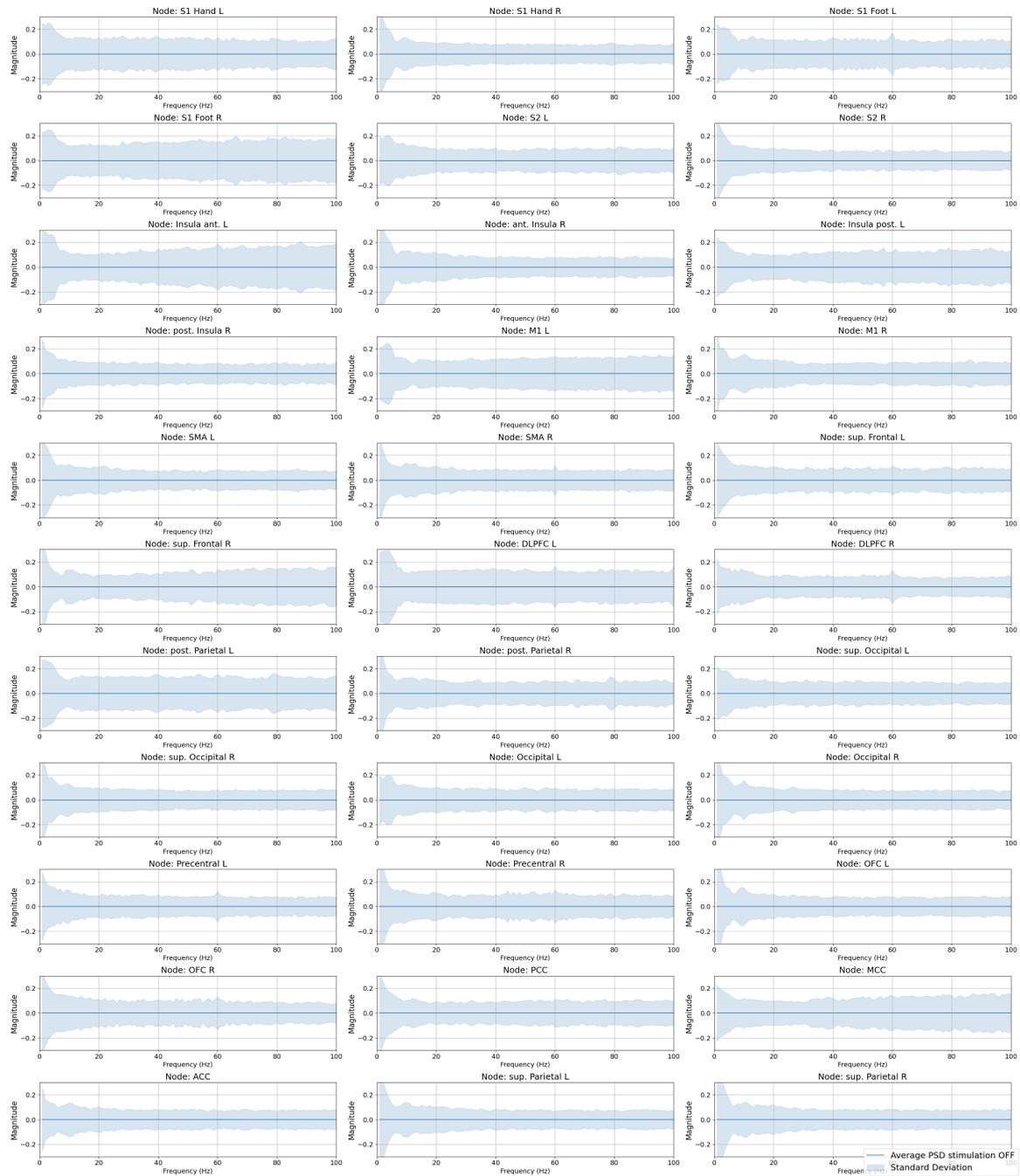
**Figure L.2:** The averaged node features for all nodes for the stimulation OFF subepochs of the sensor-space analysis.



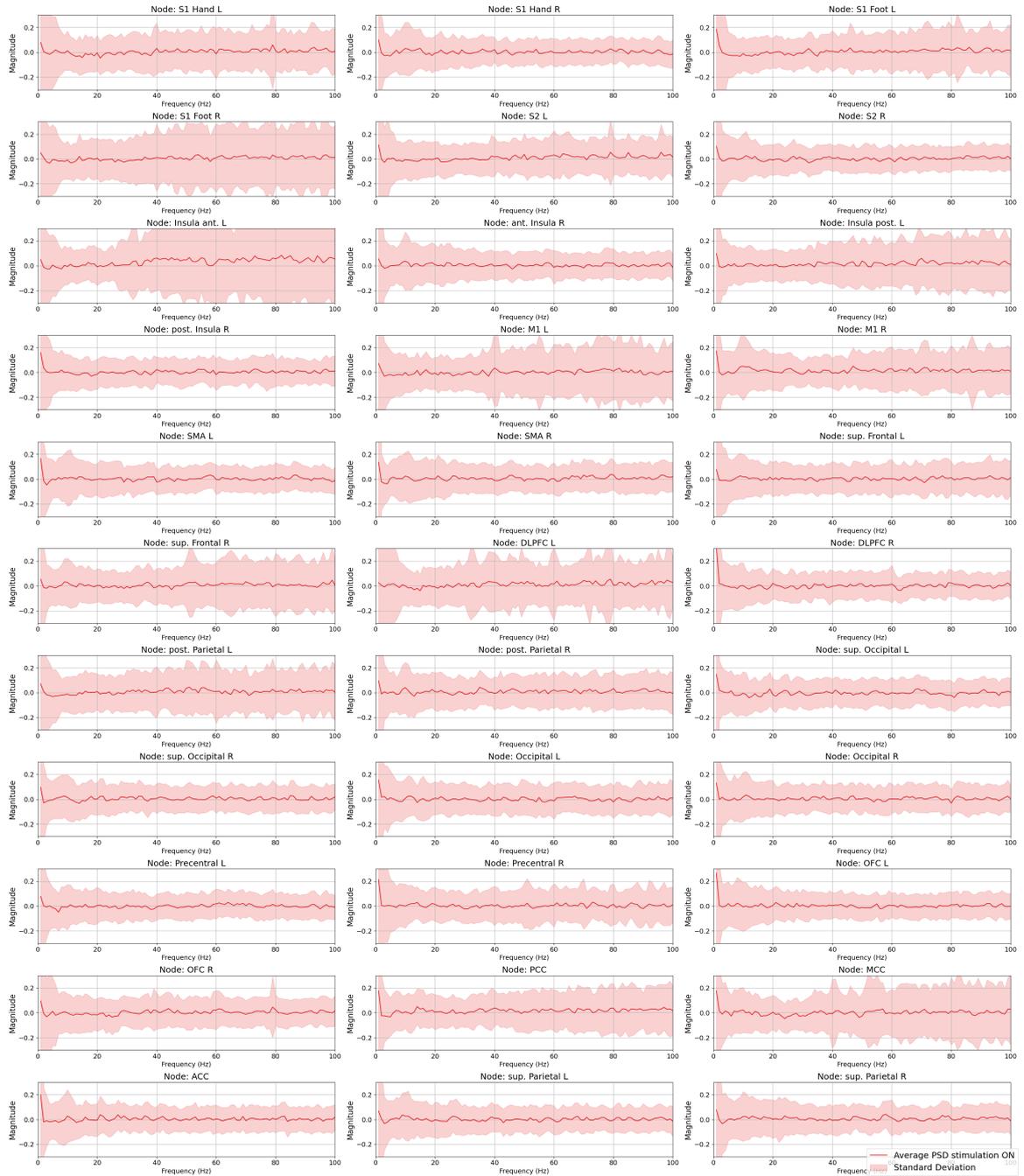
**Figure L.3:** The averaged node features for all nodes for the stimulation ON subepochs of the sensor-space analysis.



**Figure L.4:** The difference in connectivity between stimulation ON and OFF, respectively, shown as a heatmap where red indicates higher connectivity during stimulation ON, and blue indicates higher connectivity during stimulation OFF.

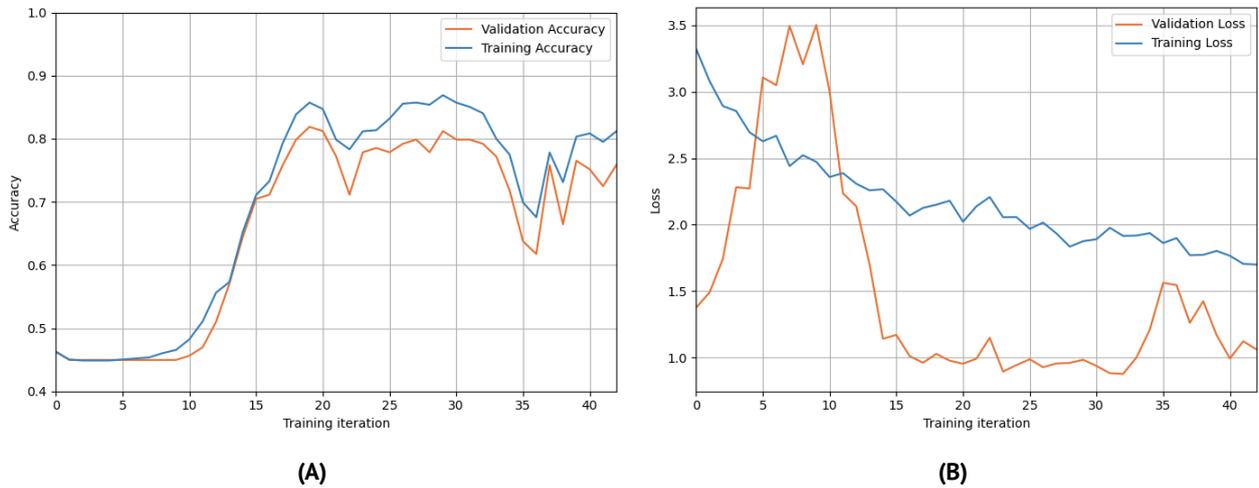


**Figure L.5:** The averaged node features for all nodes for the stimulation OFF subepochs of the source-space analysis.

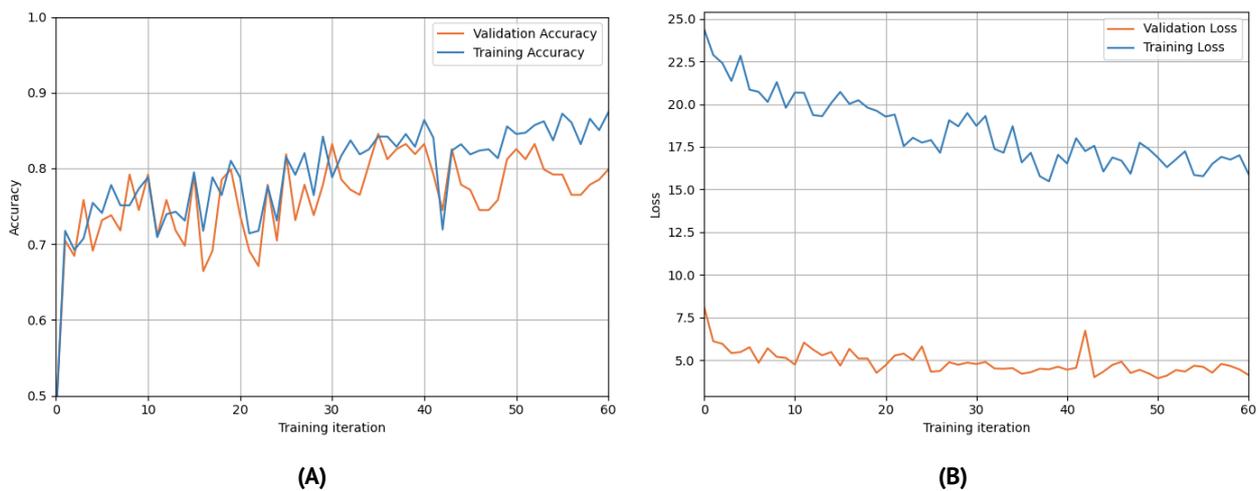


**Figure L.6:** The averaged node features for all nodes for the stimulation ON subepochs of the source-space analysis.

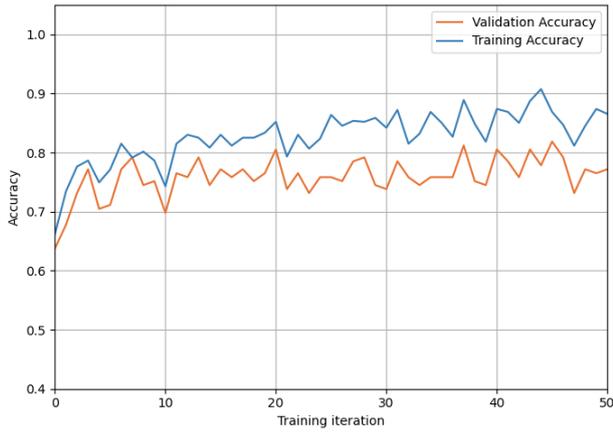
# M | Accuracy and Loss curves



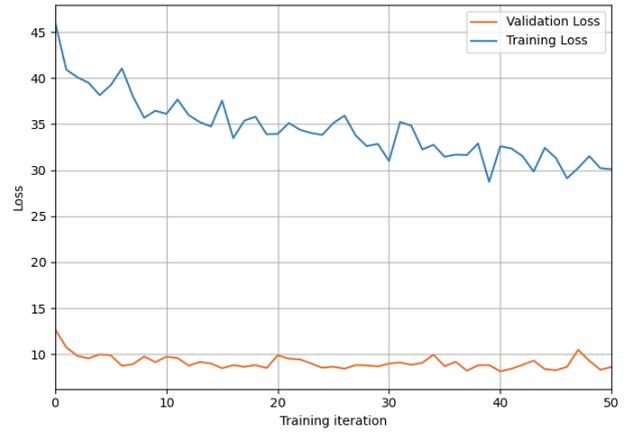
**Figure M.1:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based Graph Neural Network (GNN) model for the delta band.



**Figure M.2:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model for the theta band.

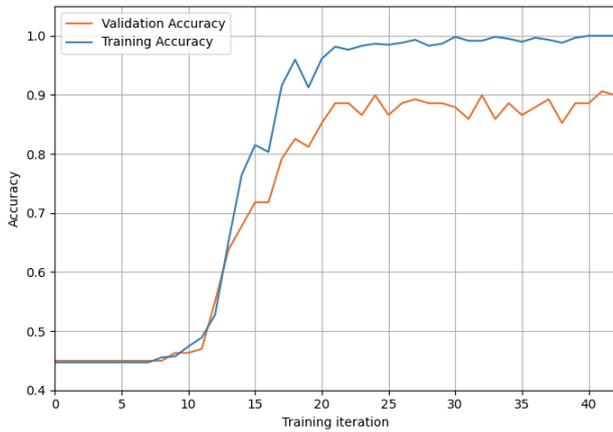


(A)

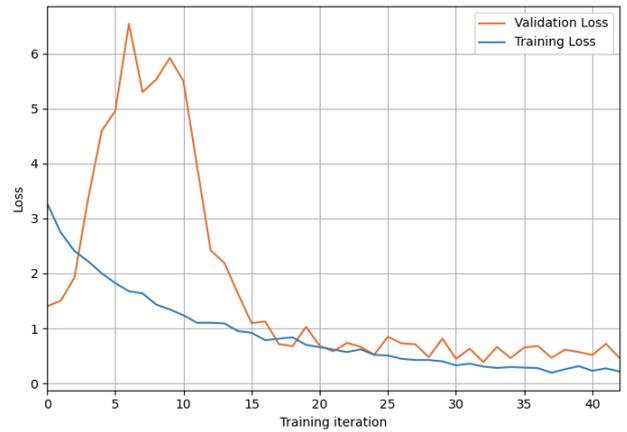


(B)

**Figure M.3:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model for the alpha band.

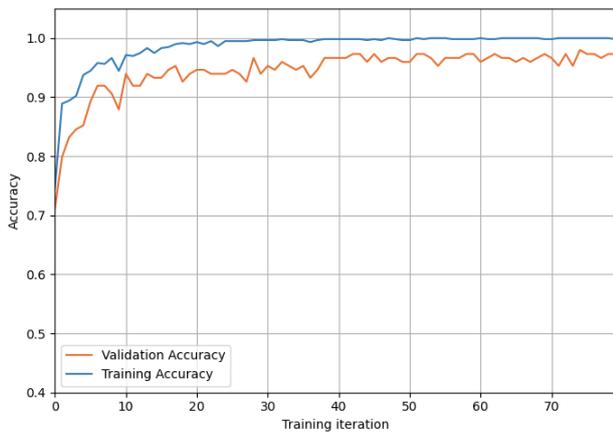


(A)

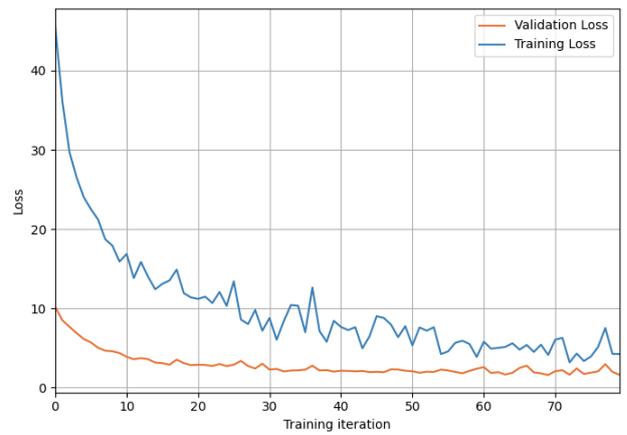


(B)

**Figure M.4:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model for the theta and alpha band.

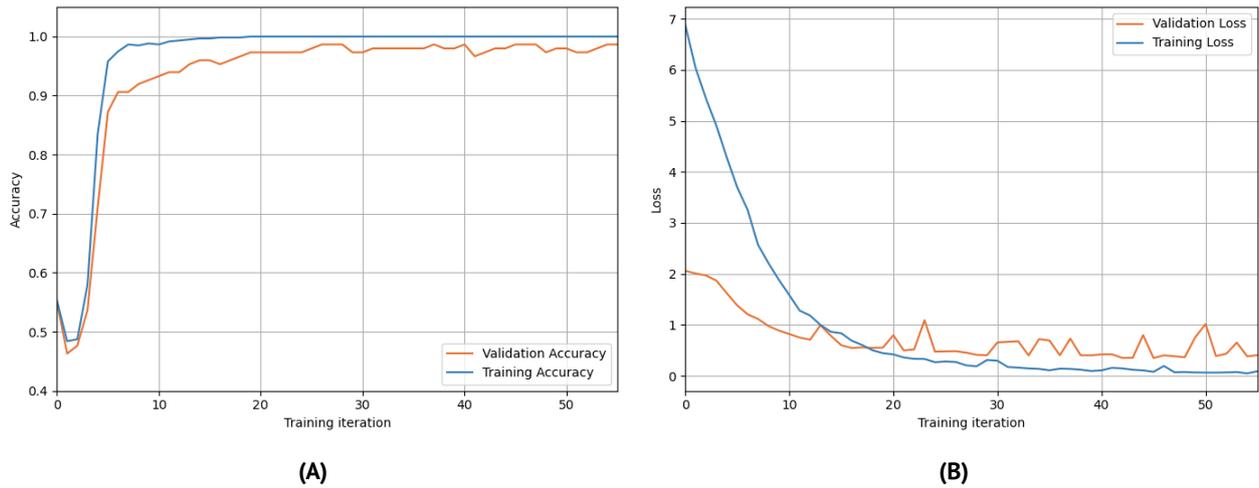


(A)

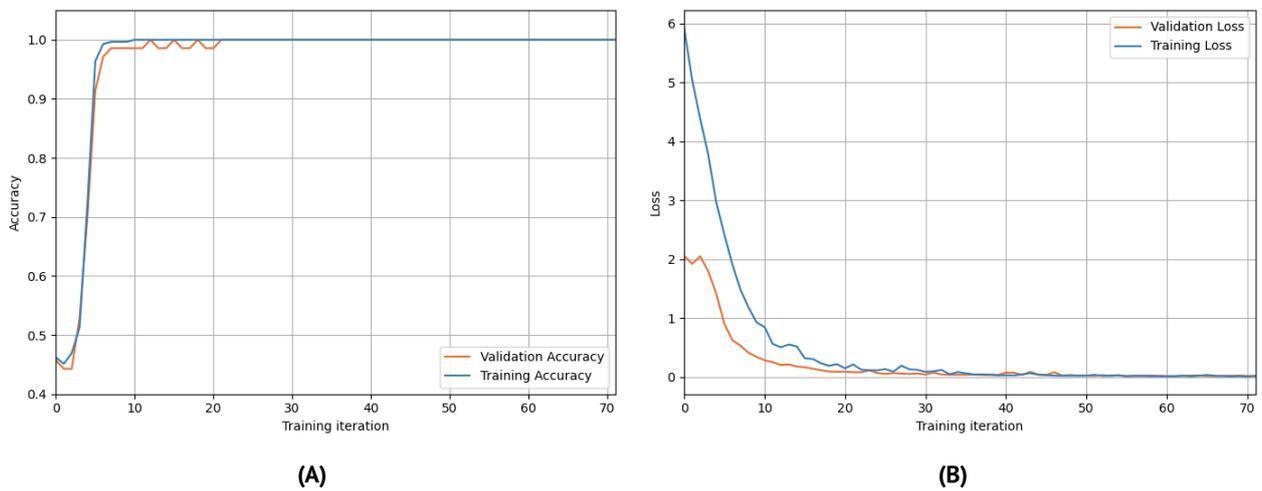


(B)

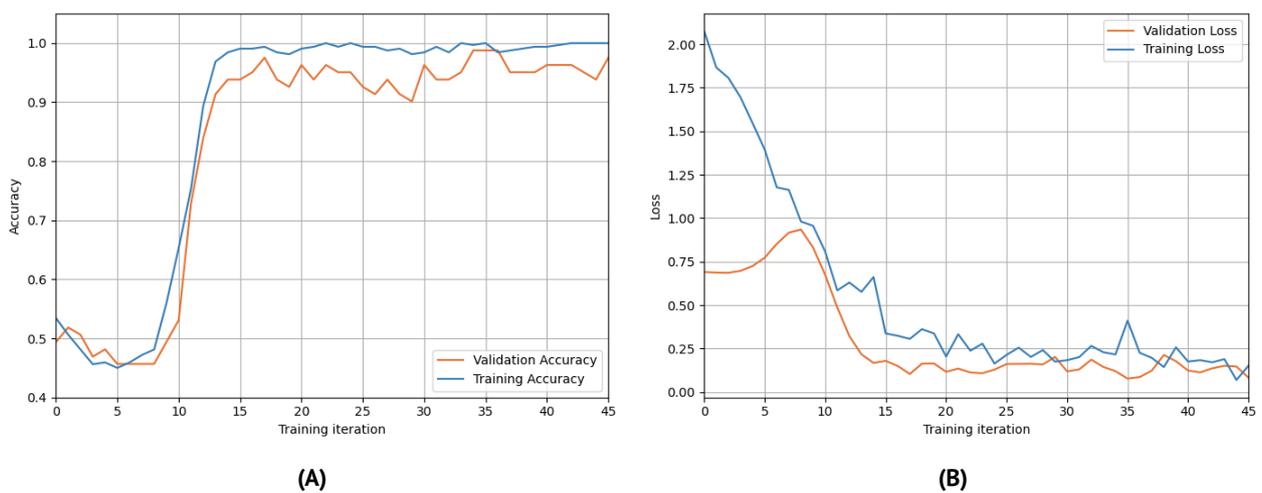
**Figure M.5:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model for the beta band.



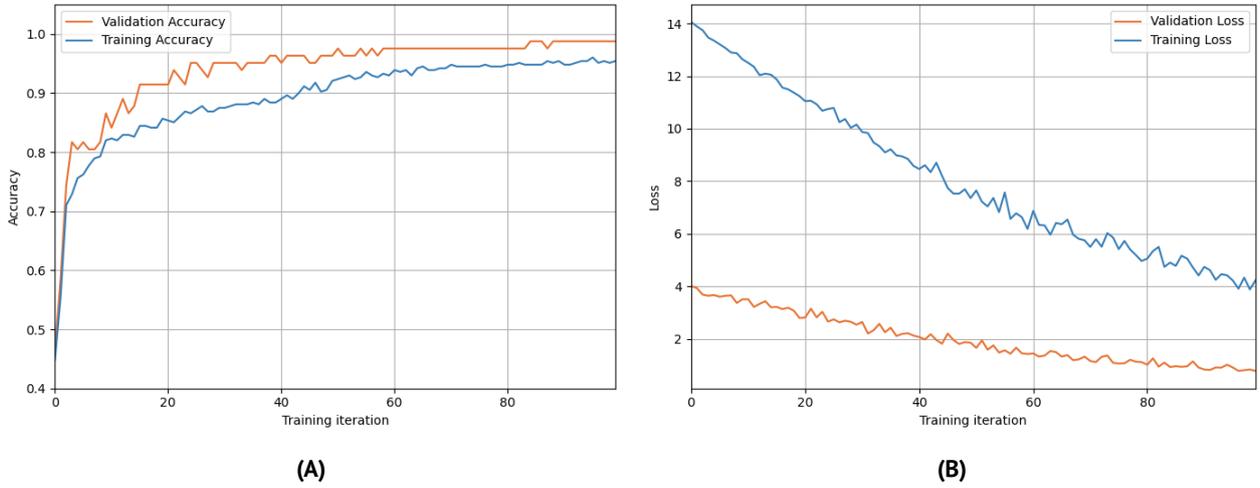
**Figure M.6:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model for the gamma band.



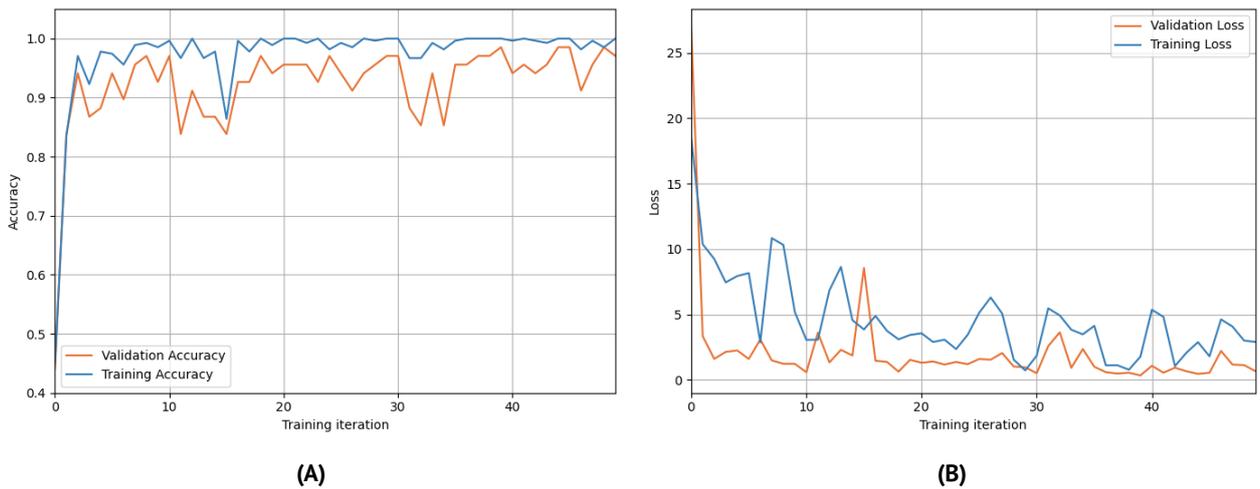
**Figure M.7:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model on the recordings using tonic Spinal Cord Stimulation (SCS).



**Figure M.8:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model on the recordings using burst SCS.

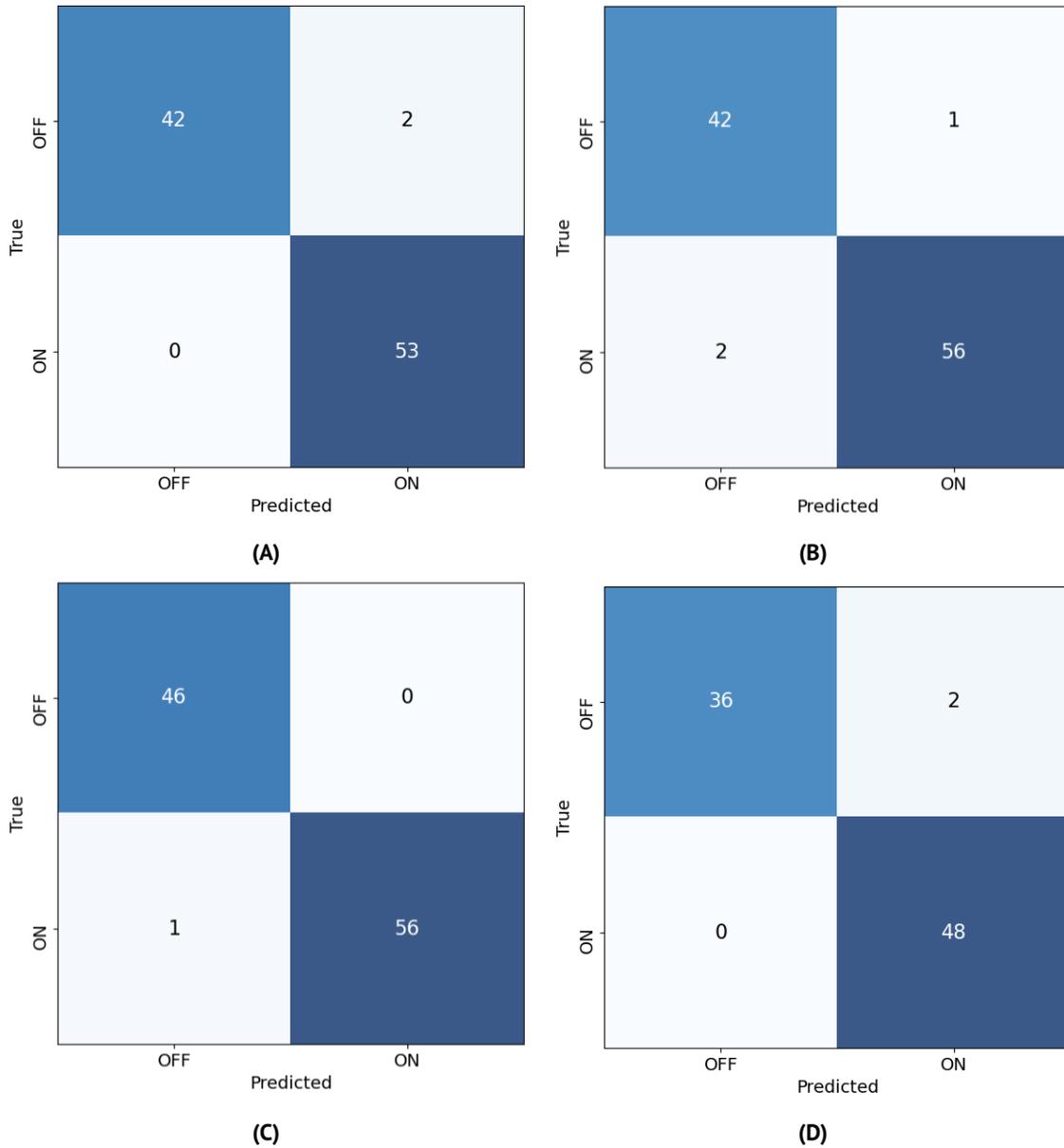


**Figure M.9:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model on the recordings from the Montreal Neurological Institute (MNI).

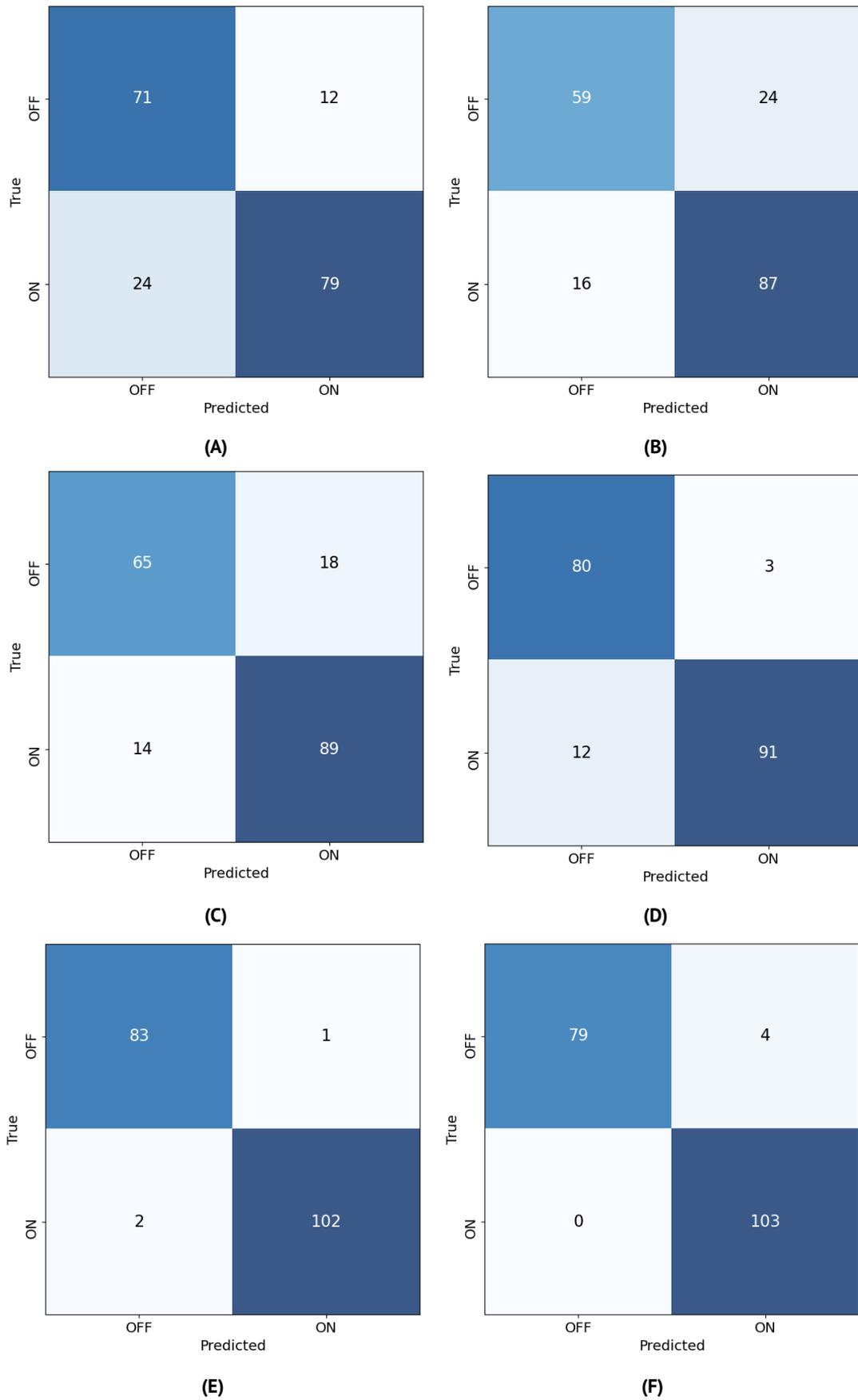


**Figure M.10:** Validation and test accuracy (A) and loss (B) over the training iterations for the sensor-based GNN model on the recordings from the Donders Institute.

# N | Confusion Matrices



**Figure N.1:** Confusion matrix of test sets for sensor-based Graph Neural Network (GNN) models trained and tested on the different stimulation paradigms, (A) tonic Spinal Cord Stimulation (SCS) and B burst SCS, and on recordings from the recording institutes (c) Montreal Neurological Institute (MNI) and (D) the Donders Institute.



**Figure N.2:** Confusion matrix of test sets for sensor-based GNN models trained and tested on the different frequency bands; (A) delta band, (B) theta band, (C) alpha band, (D) theta and alpha band, (E) beta band, and (F) gamma band.